

UNIVERZA V LJUBLJANI  
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

METODA KRANJC

UNIVERZA V LJUBLJANI  
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

IZGRADNJA PODATKOVNEGA SKLADIŠČA ZA POTREBE  
ZAVODA ZA ZDRAVSTVENO ZAVAROVANJE SLOVENIJE

Ljubljana, maj 2009

METODA KRANJC

## **IZJAVA**

Študentka Metoda Kranjc izjavljam, da sem avtorica tega magistrskega dela, ki sem ga napisala pod mentorstvom prof. dr. Vladislava Rajkoviča in skladno s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovolim objavo magistrskega dela na spletnih straneh fakultete.

V Ljubljani, dne 18.5.2009

Podpis: \_\_\_\_\_

## KAZALO VSEBINE

Uvod.....	1
<b>1 Namen in cilj .....</b>	<b>3</b>
<b>2 Metoda dela.....</b>	<b>4</b>
<b>3 Kaj je poslovna inteligenca?.....</b>	<b>4</b>
3.1 Vpeljava poslovne inteligence v poslovanje .....	5
<b>4 Podatkovna skladišča.....</b>	<b>6</b>
4.1 Poslovno obveščanje.....	8
4.2 OLAP – sprotna analitična obdelava podatkov .....	10
4.3 Podatkovno skladišče in področno podatkovno skladišče.....	10
4.4 Arhitekture podatkovnih skladišč .....	11
4.4.1 Centralizirana arhitektura.....	12
4.4.2 Distribuirana arhitektura podatkovnega skladišča .....	13
4.4.3 Federativna arhitektura podatkovnega skladišča.....	14
4.4.4 Skupni informacijski (poslovni) model.....	15
4.5 Lastnosti in razlike med OLTP in OLAP .....	16
4.6 Razlogi za neuspeh podatkovnih skladišč .....	18
<b>5 Podatkovno rudarjenje .....</b>	<b>19</b>
<b>6 Razvoj podatkovnega skladišča .....</b>	<b>20</b>
6.1 Načrtovanje podatkovnega skladišča .....	20
6.2 Življenjski cikel podatkovnega skladišča.....	22
6.3 Analiza zahtev za snovanje podatkovnega skladišča .....	23
6.4 Dimenzijsko modeliranje.....	24
6.4.1 Kaj je dimenzijsko modeliranje?.....	24
6.4.2 Tipi in uporaba dimenzijskih modelov.....	25
6.4.3 Načrtovanje večdimenzijskega modela.....	29
6.4.4 Počasi spreminjajoče dimenzije .....	31
6.4.5 Kako v zasnovi modela upoštevati performance sistema?.....	32
<b>7 Polnjenje podatkovnega skladišča .....</b>	<b>33</b>
7.1 ETL proces.....	33
7.2 Problemi pri polnjenju podatkovnih skladišč .....	34
7.3 Načini polnjenja podatkovnih skladišč .....	34
<b>8 Orodja poslovne inteligence .....</b>	<b>35</b>
<b>9 Orodje Microstrategy za delo nad podatkovnimi skladišči .....</b>	<b>36</b>
9.1 Paleta izdelkov, ki sestavlja Microstrategy platformo.....	36
9.2 Opis funkcionalnosti orodja .....	37
9.3 Dinamične nadzorne plošče .....	38
<b>10 Kratka predstavitev Zavoda za zdravstveno zavarovanje Slovenije ....</b>	<b>39</b>
10.1 Poslanstvo in vizija Zavoda .....	39
10.2 Vizija razvoja informacijskega centra .....	42
10.3 Organizacijska struktura Zavoda.....	45
10.4 Kratek opis direktorskega informacijskega sistema.....	45
10.5 Cilji zagotavljanja podpore na področju podatkovnega skladiščenja v Zavodu	45
<b>11 Izgradnja podatkovnega skladišča v Zavodu za primer projekta STIRA</b> <b>(Predpisovanje zdravniških receptov po izvajalcih in zdravnikih glede na</b> <b>dejavnost) .....</b>	<b>46</b>
11.1 Kratek opis področja STIRA (predpisovanje receptov po zdravnikih).....	46

<b>11.2. Izhodišča in kriteriji za izračun ciljnih vrednosti po metodologiji STIRA .....</b>	<b>47</b>
11.2.1 Metodologija ciljnih vrednosti .....	47
11.2.2 Obdelava podatkov in možna odstopanja.....	49
11.2.3 Število in vrednost receptov ter indeksi odstopanja od ciljnih vrednosti za recepte po območnih enotah v določenem letu.....	50
11.2.4 Proces obdelave in logistika podatkov STIRA .....	50
11.2.5 Primerjava obstoječega in novega načina posredovanja podatkov STIRA.....	51
<b>11.3 Načrtovanje podatkovnega skladišča za projekt STIRA.....</b>	<b>52</b>
11.3.1 Organizacija dela.....	52
11.3.2. Potrebni koraki za vzpostavitev sistema .....	54
11.3.3 Opis arhitekture sistema .....	55
11.3.4 Analiza in zbiranje zahtev uporabnikov .....	56
11.3.5 Načrtovanje dimenzijskega modela .....	58
11.3.6 Vsebina procesa polnjenja podatkovnega skladišča.....	59
11.3.7 Izbira tehnologije oziroma načina polnjenja podatkovnega skladišča.....	61
<b>11.3.8 Polnjenje podatkovnega skladišča s pomočjo orodja DataStage .....</b>	<b>62</b>
11.3.9 Implementacija analiz z orodjem Microstrategy ter kratka predstavitev uporabe orodja.....	65
11.3.10 Testiranje delujoče rešitve.....	67
11.3.11 Dokumentiranje v procesu načrtovanja podatkovnega skladišča.....	67
11.3.12 Izobraževanje uporabnikov za delo z orodjem Microstrategy .....	68
11.3.13 Vzdrževanje delujoče rešitve .....	68
<b>Zaključek.....</b>	<b>69</b>
<b>12 Literatura in viri.....</b>	<b>71</b>
<b>13 PRILOGE.....</b>	<b>1</b>
13.1 Struktura posameznih tabel za projekt STIRA .....	1
13.2 Kratek opis uporabe orodja Microstrategy .....	4
13.3 Kratek opis uporabe orodja DataStage na primeru projekta STIRA .....	9
14.4 SQL-i za preverjanje in polnjenje podatkovnega skladišča STIRA.....	14
13.5 Organigram za direkcijo in poslovno enoto informacijski center. ....	15

## KAZALO SLIK

<i>Slika 1: Mesto in vloga podatkovnega skladišča v informacijsko upravljalskem procesu poslovnega sistema</i> .....	8
<i>Slika 2: Nov potencial poslovnega obveščanja</i> .....	8
<i>Slika 3: Arhitektura centraliziranega podatkovnega skladišča</i> .....	13
<i>Slika 4: Distribuirano podatkovno skladišče</i> .....	14
<i>Slika 5: Federativna arhitektura podatkovnega skladišča</i> .....	15
<i>Slika 6: Faze razvoja življenjskega cikla</i> .....	23
<i>Slika 7: Shema Zvezde in Snežinke</i> .....	26
<i>Slika 8: Vhodi za dimenzijsko modeliranje</i> .....	30
<i>Slika 9: Primer dinamičnih nadzornih plošč</i> .....	39
<i>Slika 10: Arhitektura sistema projekta STIRA</i> .....	56
<i>Slika 11: Dimenzijski model za projekt STIRA</i> .....	59
<i>Slika 12: Prikaz načina polnjenja podatkovnega skladišča</i> .....	60
<i>Slika 13: Prikaz umestitve orodja DataStage ter povezava z Microstrategy</i> .....	63
<i>Slika 14: Sequence job oz. control flow (kontrolni tok) za povezovanje posameznih podatkovnih procesov med seboj</i> .....	64
<i>Slika 15: Kreiranje Project Sourca in Projekta ter prikaz objektov projekta</i> .....	66
<i>Slika 16: Primer kreiranja hierarhije</i> .....	66
<i>Slika 17: Kreiranje filtra za indeks &gt; 100</i> .....	67

## KAZALO TABEL

<i>Tabela 1: Razlike med OLTP in OLAP</i> .....	17
<i>Tabela 2: Primerjava in tipi shem</i> .....	28
<i>Tabela 3: Primer matrike po Kimball-u</i> .....	31
<i>Tabela 4: Moduli orodja Microstrategy</i> .....	37
<i>Tabela 5: Primer Prikaz izračuna količnikov za osnovno zdravstveno dejavnost za Slovenijo na podlagi podatkov za določeno leto</i> .....	48
<i>Tabela 6: Skrbništvo sistema orodja Microstrategy</i> .....	53
<i>Tabela 7: Vloge pri izvajanju nalog za projekt STIRA</i> .....	53
<i>Tabela 8: Struktura poročil za projekt STIRA – prikaz podatkov po izvajalcih in zdravnikih glede na dejavnost</i> .....	57
<i>Tabela 9: Struktura poročil za projekt STIRA – prikaz podatkov po izvajalcih glede na dejavnost</i> .....	57
<i>Tabela 10: DataStage odjemalci</i> .....	63



## Uvod

Zavod za zdravstveno zavarovanje Slovenije (v nadaljevanju: Zavod) je bil ustanovljen 1. marca 1992. Zavod izvaja obvezno zdravstveno zavarovanje s prispevki, ki jih plačujejo z zakonom opredeljeni zavezanci. V začetku leta 1997 je Zavod na osnovi obsežnih analiz pripravil nov razvojni dokument - Strateški razvojni program zdravstvenega zavarovanja v Sloveniji. V letu 2002 je bil izdelan in sprejet nov strateški razvojni program z naslovom "Tradicija solidarnosti in spremembe po meri zavarovancev". Omenjen program podaja strateške usmeritve, cilje in projekte Zavoda za obdobje do leta 2007, ki so nujni za dograjevanje sistema obveznega zdravstvenega zavarovanja. Osrednja značilnost tega razvojnega programa je postavitve zavarovanca v središče razvojnih prizadevanj. V letu 2006 je skupščina sprejela dopolnitve Strateškega razvojnega programa Zavoda za leti 2006 in 2007, ki je v elektronski obliki dostopno tudi preko spletnih strani. Zavod je od sprejema prvega razvojnega programa pa do danes izvedel številne razvojne aktivnosti in projekte, med katerimi velja z vidika razvoja Zavoda in zdravstvenega zavarovanja nekatere še posebej izpostaviti: V letu 2000 je Zavod dokončno uvedel sodobno elektronsko kartico zdravstvenega zavarovanja (KZZ) v slovensko zdravstvo. V obdobju 2002-2006 so bili na tem področju izvedeni še nekateri projekti, ki so omogočili nove uporabnosti kartice – zapis izdanih zdravil, medicinsko tehničnih pripomočkov ter podatkov o darovalcu organov na kartico. Zaradi številnih poslovnih in tehničnih razlogov se je Zavod v letu 2006 odločil za prenovo kaotičnega sistema, in sicer postopno v smeri uvedbe neposrednih dostopov do podatkov, ki se sedaj nahajajo na kartici. Enostavno, kakovostno in učinkovito prenašanje podatkov ter komuniciranje v okviru on-line zdravstvenega zavarovanja je namreč ena temeljnih razvojnih usmeritev Zavoda v prihodnjih dveh letih, ki jo bo Zavod skušal uresničiti z izvedbo dveh projektov, ki bosta usmerjena v uvedbo on-line sistema ter razvoj nove kartice zdravstvenega zavarovanja, profesionalne kartice in infrastrukture javnih ključev. S tem se bodo za zavarovane osebe poenostavili postopki uresničevanja pravic iz zdravstvenega zavarovanja, izvajalcem zdravstvenih storitev bodo na voljo bolj točni in ažurni osebni in medicinski podatki, kar bo povečalo kakovost zdravstvenih storitev, zdravstvenim zavarovalnicam pa omogočilo racionalizacijo stroškov poslovanja.

Omenjenih je bil le nekaj glavnih informacij o informacijskih projektih Zavoda za zdravstveno zavarovanje in kartici zdravstvenega zavarovanja.

Institucija kot je Zavod za zdravstveno zavarovanje Slovenije tekom dneva pridobiva in izmenjuje pomembne informacije o zavarovancih z izvajalci zdravstvenih storitev ter z drugimi zunanjimi institucijami. Z denarnimi sredstvi, ki jih ima na razpolago mora ravnati racionalno in preudarno ter slediti strateškemu razvoju Zavoda. In prav podrobne analize, ki jih lahko omogočimo s pomočjo podatkovnih skladišč prispevajo k učinkovitosti posameznih procesov. Danes je informatika znanost od katere je odvisen ves svet, zato jo je treba temu potrebno tudi obravnavati (Krajnc Pavlica, 2005, str. 1).

Poslovna inteligenca je postala kritični element informacijske tehnologije. Izraz se je v preteklosti uporabljal kot sinonim za podporo procesom odločanja in podatkovnim skladiščem. Danes ima pojem poslovna inteligenca bolj specifičen pomen in boljše razumevanje v poslovanju, saj je bolj uspešno in učinkovito implementirana. Na kratko je poslovna inteligenca le razumevanje poslovnih procesov, razumevanje pa pridobimo le s pomočjo analiziranja poslovnih operacij. Platforma poslovne inteligence naj bi bila integracija tehnologij kot so: baze podatkovnih skladišč, olap, podatkovno rudarjenje, dostopni uporabniški vmesniki do baz podatkov, zmožnosti zgraditi, vzdrževati ter polniti podatkovna



skladišča. Za prikaz analiz si lahko pomagamo z različnimi programskimi jeziki ali močnimi vizualnimi orodji, ki obstajajo na tržišču. (Hill & Kramer, 2002, str. 1-3).

Pomembnost poslovne inteligence se vedno bolj poudarja tudi na raznih konferencah in posvetovanjih. Poslovna inteligenca nudi celosten pogled na podjetje: spremlja in omogoča vpogled v finančno in operativno zdravje organizacije, pomaga pri upravljanju delovanja organizacije, z hitrim zaznavanjem trendov, podjetju omogoča, da je korak pred konkurenco (Hudoklin, 2004, str. 6).

Ralph Kimball (1996) označuje podatkovno skladišče kot prostor, kjer lahko ljudje dostopajo do svojih podatkov. Podatkovno skladišče torej predstavlja shrambo podatkov, ki pa morajo biti na voljo uporabnikom.

Barry Devlin (1997) daje poudarek kakovosti in konsistentnosti podatkov v podatkovnih skladiščih. Podatkovno skladišče opisuje kot: »enotno, popolno in konsistentno hrambo podatkov, pridobljenih iz različnih virov in ponujenih končnim uporabnikom na način, ki ga razumejo in uporabljajo v poslovnem svetu.

Skladišče podatkov torej omogoča, da se podatki različnih sistemov združijo v enotno zbirko podatkov, se obravnavajo na enoten način in so dostopni vsem pooblaščenim uporabnikom. Hranijo se lahko tudi zgodovinski podatki zaradi pregleda po časovnih vrstah, česar transakcijski sistemi običajno ne omogočajo. V skladišče podatkov je mogoče prenesti razen podatkov in poslovnih informacijskih sistemov podjetja tudi podatke, pridobljene iz zunanjih virov. Proces prenosa podatkov v skladišče imenujemo s tujo kratico ETL – Extract Transform Load. ETL proces pa še zdaleč ni le enostaven proces polnjenja dimenzijskih tabel, temveč kot pravi Ralph Kimball, so etl procesi v življenjskem ciklu izgradnje podatkovnega skladišča delež najtežji, časovno zahtevni in delovno intenzivni (Kimball, 2004).

V skladišču podatkov so podatki oblikovani na način, ki omogoča uporabnikom ali razvijalcem preprosto izdelavo poročil in analiz podatkov, saj so podatki shranjeni v obliki dimenzij in mer. Danes razvijajoča se informacijska tehnologija ponuja veliko zmogljivih orodij za delo s podatki v podatkovnih skladiščih. Delo je postalo učinkovitejše, saj je uporabniku omogočeno, da lahko sami izdelajo poročila, kadar jih potrebujejo in jim ni potrebno čakati, da jim jih izdelata kdo drug (SRC, 2006). V Zavodu se uporablja eno izmed petih najboljših orodij na tržišču – Microstrategy.

Že pri sami zasnovi podatkovnega skladišča je pomembno, da se razvijalci odločijo, kakšna bo arhitektura in čemu bo služilo. Tako ločimo celovita, specializirana in kombinirana podatkovna skladišča (R. Kimball, študijsko gradivo, 2006), drugi avtorji jih poimenujejo kot centralizirana, distribuirana in federativna (Golob & Welzer, 2001).

Seveda je izgradnja podatkovnega skladišča med izvajanjem nagnjena k številnim tveganjem. Največja težava je v tem, da med izvajanjem ni mogoče oz. je otežkočen hiter prikaz rezultata vlaganj v izgradnjo podatkovnega skladišča. Če se le da, naj se managementu pokaže otipljive izdelke oz. rezultate vsake tri mesece. Če otipljivih rezultatov ni, se s tem začne zmanjševati zaupanje vodstva in drugih zaposlenih, nenazadnje pa tudi neposredno vključenega osebja v proces izgradnje podatkovnega skladišča, kar posledično pomeni veliko verjetnost neuspeha oz. prekoračitev tako časovnih kot finančnih okvirov samega procesa izgradnje (Berce, 2001, str. 10).

V Zavodu so leta 1999 že zaključili projekt uvajanja celovitega podatkovnega skladišča oziroma direktorskega informacijskega sistema (DIS). Kasneje se sistem ni izkazal kot za zelo uspešnega, saj so se začeli pojavljati razni problemi. V tistem času je DIS uvajalo zunanje podjetje, notranje znanje glede področja podatkovnih skladišč pa je bilo zelo slabo. Tako se je pred dvema letoma porodila ideja o ustanovitvi nove notranje ekipe za omenjeno področje, kjer sem zaposlena tudi sama. Ekipe se ukvarjala z uvajanjem manjših podatkovnih skladišč oziroma kot jih nekateri avtorji imenujejo podatkovne tržnice. Glede na raznolikost področij zdravstvenega zavarovanja se pojavljajo vedno novi izzivi in naloge, ki jih bo potrebno analizirati, tako se ekipi tudi v prihodnje ni bati, da bi ostala brez dela. Obstoječa informacijska tehnologija pa naj bi omogočala, da bi se odločali in upravljali bolje kot doslej.

Podatkovno skladišče lahko vsebuje združene podatke iz poljubnih sistemov in predstavlja podjetju glavni temelj za analize, poročanje, planiranje ter simulacije poslovanja. V Zavodu obstaja več vrst informacijskih sistemov kot so: Lotus Notes, SAP, Main frame DB2, itd. Pojavlja se vedno več potreb po analizah in integracij podatkov iz omenjenih področij in prav s pomočjo podatkovnih skladišč se bodo lahko izvajale razne analize za podporo procesom odločanja.

Obstoječe podatkovno skladišče se je polnilo s pomočjo obsežnih PL programov. Ker naj bi se v prihodnje razvijalo vedno več ločenih podatkovnih tržnic, procese polnjenja pa bi morali imeti avtomatizirane, se je ekipi porodila tudi ideja o testiranju nekaterih ETL orodij. Že zgoraj je bilo omenjeno, da je ETL pomemben in zahteven proces, saj je od njega odvisna kvaliteta in pravilnost podatkov v podatkovnem skladišču. Omenjena orodja naj bi poleg avtomatizacije postopkov omogočala tudi dobro dokumentacijo procesov polnjenja.

## **1 Namen in cilj**

V magistrski nalogi sem si zadala več ciljev. V prvih poglavjih bo predvsem poudarek na teoretičnem poznavanju področja podatkovnih skladišč in same poslovne inteligence.

V nadaljevanju bodo predstavljene prednosti in slabosti med uvajanjem celovitega podatkovnega skladišča ter področnih tržnic. Ekipe je poskusila obdržati prednosti in koristi preteklega uvajanja podatkovnega skladišča ter s pomočjo pridobivanja novih znanj prispevati k uspešnejši izgradnji novih podatkovnih tržnic.

V magistrski nalogi bo opisan praktični primer izgradnje podatkovnega skladišča za obstoječi projekt STIRA oz. predpisovanja zdravniških receptov po izvajalcih in zdravnikih glede na dejavnosti. Opisane bodo vse faze izgradnje podatkovnega skladišča od analize zahtev, dimenzijskega modeliranja, načrtovanje rešitve z orodjem Microstrategy, polnjenja podatkovnega skladišča ter do vzdrževanja in izobraževanja uporabnikov. Projekt je bil glede na zahteve končan spomladi leta 2008 leta. Obsežno šolanje za večje število uporabnikov se je izvedlo avgusta 2008. So se pa že pojavili nove zahteve pa nadgradnji projekta STIRA. Cilj magistrskega dela bo tako podati prednosti in pomanjkljivosti med starim in novim načinom dela.

K učinkovitemu uvajanju podatkovnih skladišč pripomore tudi ustrezna informacijska podpora. Mišljena je informacijska tehnologija, ki jo potrebujemo pri polnjenju podatkovnih skladišč, hranjenju podatkov ter do orodij poslovne inteligence, ki nam pomagajo do raznih analiz in poročil nad podatki podatkovnega skladišča. Tako bo v magistrski nalogi poleg

splošnega oz. teoretičnega opisa tehnologij, opisana tehnologija, ki se je uporabila pri omenjenem projektu STIRA.

Cilj bo tudi na podlagi primera opisati potek in probleme pri polnjenju podatkovnega skladišča ter podati izsledke testiranja nekaterih ETL orodij in predlagati najboljši možni način za polnjenje podatkovnih tržnic. V Zavodu so se odločili za nakup ETL orodja DataStage, katerega delovanje bo na kratko predstavljeno tudi v magistrski nalogi.

## 2 Metoda dela

V magistrski nalogi bom uporabila domačo in tujo strokovno literaturo, predvsem pa svoja znanja in izkušnje s področja izgradnje podatkovnih skladišč.

Celotna magistrska naloga bo v grobem razdeljena na 3 dele. V prvem delu bodo podana teoretična izhodišča, ki bodo temeljila na podlagi študija strokovne literature domačih in tujih avtorjev, na virih, aktualni prispevkih in člankih s področja podatkovnih skladišč in poslovne inteligence. V uvodu bo predstavljen tudi sam namen in cilj dela. V poglavju bodo vključene vsebine s področja podatkovnih skladišč, podatkovnega rudarjenja, dimenzijskega modeliranja, orodij poslovne inteligence, itd..

V drugem delu bo predstavljena institucija Zavoda za zdravstveno zavarovanje Slovenije, njegova organizacijska struktura, vizija in poslanstvo. Na kratko bo opisan tudi obstoječi direktorski informacijski sistem ter cilji zagotavljanja podpore na področju podatkovnega skladiščenja v Zavodu.

V tretjem delu bo predstavljen praktični primer uvajanja manjšega področnega skladišča v Zavodu. Omenjeno je bilo, da gre za projekt STIRA oziroma projekt predpisovanja zdravniških receptov po izvajalcih in zdravnikih glede na zdravstveno dejavnost. Opisane bodo vse faze uvajanja področnega skladišča od začetne do končne stopnje, saj nobene faze ni za zanemariti za uspešno uvedbo le tega v praksi. Se pravi opisani bodo koraki od zajema in analize zahtev, do dimenzijskega modeliranja, implementacije rešitve, polnjenja podatkovnega skladišča, dokumentiranja ter vzdrževanja in izobraževanja uporabnikov.

Na koncu bodo podani tudi napotki za nadaljnje delo.

## 3 Kaj je poslovna inteligenca?

Poslovno inteligenco ali kot jo s kratico označujemo BI (angl. *Business intelligence*) nekateri imenujejo tudi nova priložnost za izboljšanje uspešnosti podjetij ali sistem za lažje in boljše odločanje. Vsi veliki proizvajalci imajo danes bolj ali manj podobne rešitve, katere ponujajo ali kot samostojne rešitve ali kot dele svojih že uveljavljenih ERP (angl. *Enterprise Resource Planing*) paketov oziroma rešitev.

V strokovnih literaturah lahko najdemo kar nekaj definicij poslovne inteligence:

- poslovna inteligenca je poslovno ravnateljski (managementski) izraz, ki opisuje aplikacije in tehnologije, ki se uporabljajo za zbiranje, dostop do in analizo podatkov in informacij o delovanju podjetij. Je odgovor informacijske industrije

na potrebe podjetij po hitrem odločanju, ki sloni na pred definiranih kazalcih, vzorcih ali drugih pravilih, ki jih podjetja uporabljajo za svoje odločanje (wikipedia, 2009);

- je sistem, ki v povezavi s podatkovnim skladiščem omogoča podjetju, da analizira pretekle dogodke z namenom razumevanja trenutnega položaja podjetja ter priprave bodočih trendov razvoja (Imhoff et al, 2003, str 4.);
- je splošni izraz, ki opisuje uporabo notranjih in zunanjih informacijskih virov podjetja na način, ki omogoča lažje in boljše odločitve (Kimball, 2002, str 393);
- poslovna inteligenca nudi celosten pogled na podjetje: spremlja in omogoča vpogled v finančno in operativno zdravje organizacije, pomaga pri upravljanju delovanja organizacije, z hitrim zaznavanjem trendov, podjetju omogoča, da je korak pred konkurenco (Hudoklin, 2004).

BI naj bi pomagala podjetjem pridobiti bolj poglobljena znanja o zadevah, ki vplivajo na njihovo poslovanje, kot so to različni kazalci prodaje, proizvodnje, internega poslovanja itn. in jim omogočajo boljše poslovno odločanje. Čar BI orodij oziroma sistemov je, da uporabljajo tako strukturirane kot nestrukturirane vire podatkov oziroma informacij. Viri za BI so lahko naša obstoječa podatkovna skladišča, transakcijski sistemi, naši notranji oziroma zunanji portali, naši ERP sistemi, informacijski sistemi naših dobaviteljev, javno dostopne baze, sistemi za upravljanje z dokumenti itd.

BI aplikacije se uporabljajo na različnih področjih, z njimi lahko analiziramo uspešnost projektov, performance podjetja ali posamezne poslovne aktivnosti, kot je to poslovno planiranje, »performance management«, nabavno verigo, itn.

### **3.1 Vpeljava poslovne inteligence v poslovanje**

Kot pri vseh podobnih projektih je pomembna definicija cilja, ki ga želimo doseči z BI-jem. Ko imamo cilj, se moramo zavedati, da kljub temu, da bomo BI vpeljali v podjetje s projektnim načinom dela, je BI dnevna naloga oziroma proces in ga moramo, če želimo od njega dobiti ustrezno dodano vrednost, vestno izvajati. Naprej se moramo zavedati štirih V-jev, ki se nanašajo na karakteristike podatkov, ki jih analizirajo BI rešitve in to so: Volume (obseg), Velocity (hitrost), Variety (raznovrstnost) in Validity (veljavnost). Torej značilnost BI rešitev je, da so učinkovite, ko imajo na razpolago sorazmerno veliko število podatkov, da je hitrost spreminjanje teh podatkov precej višja kot na primer pri podatkovnih skladiščih, da so podatki, ki jih obravnavamo med seboj zelo različni po svojih lastnostih in da je kvaliteta rezultatov analiz BI orodij odvisna od veljavnosti oziroma nespornosti podatkov, ki jih BI uporablja.

Zelo pomembno je, da se na začetku določi posebno skupino (team), ki bo skrbelo za delovanje BI rešitve. V ta team bi bilo smiselno vključiti informatike (arhitekta informacijskih rešitev, različne specialiste, analitika kvalitete podatkov, ..) in jasno, uporabnike. Nenazadnje se je potrebno odločiti, kam vse bomo postavili oziroma kako široko bomo implementirali BI rešitev. Obstaja kar nekaj opcij, od popolne razpršenosti do zadnjega prodajnega referenta in celo naših partnerjev, do umestitev v tista poslovna področja in v roke zaposlenih, ki odločajo in ki se nam zdijo najbolj kritična. Stremeti moramo, da dosežemo tisto kar je najbolj pomembno in kar daje največjo dodano vrednost BI rešitvam in to je čim hitrejše odločanje v realnem času. Pomembno je, da ne glede na strategijo in obseg uvedbe BI rešitve začnemo z malimi koraki in jih postopoma nadgrajujemo in širimo.

Poleg cene (nekatero najbolj priznane rešitve stanejo tudi do par tisoč EUR na uporabnika, kar je samo del stroškov za implementacijo), bo za mnoga podjetja izziv že definiranje ključnih kazalcev uspešnosti (angl. *Key performance indicators* (KPIs)), s katerimi ponavadi podjetja opisujejo svoja posamezna trenutna stanja in s katerimi opisujejo zeleno pot v svojem poslovanju, kar je osnova za uvedbo BI rešitev (Hajtnik, Stajič, 2007).

## 4 Podatkovna skladišča

Hitre spremembe poslovnih procesov in tehnologij predstavljajo za sodobno podjetje bistvene organizacijske spremembe, zahtevajo daljnosežne odločitve, obenem pa pomenijo tudi izjemne priložnosti.

Vstopamo v svet relacijskih in objektnih podatkovnih skladišč in za uporabo programov uporabljamo objektno usmerjena razvojna orodja. Vse to prinaša programe nove kvalitete in bistveno večjo zanesljivost delovanja. Cena, ki jo morajo plačati podjetja in razvijalci, je zelo velika. Pravilno izbrano orodje in predvsem pravi dostop so bistvo uspeha.

Koncept podatkovnih skladišč izhaja iz potrebe lažjega dostopa do kvalitetnih podatkov, ki so shranjeni v strukturirani obliki in se lahko uporabijo za odločanje. Globalno veljajo informacije za močan doprinos, zagotavljajo velike koristi vsake organizacije in veljajo za konkurenčno prednost v poslovnem svetu. Organizacije imajo velike količine podatkov, vendar je včasih težko zajeti in jih narediti koristne. To je ponavadi zato, ker se podatki nahajajo v različnih formatih, se nahajajo na različnih platformah, datotekah, bazah, ki so lahko od različnih proizvajalcev. Te organizacije morajo napisati in vzdrževati mnogo programov za pridobivanje, pripravo in konsolidacijo podatkov, ki jih uporablja veliko aplikacij za analizo in poročanje. Ko imajo uporabniki postavljene začetne zahteve, želijo pri odločanju pogosto kopati globlje po podatkih. To pa ponavadi zahteva tudi razvoj novih ali spremembe obstoječih programov. Omenjeni procesi so neučinkoviti ter stroškovno in časovno potratni. Podatkovna skladišča nudijo boljši pristop.

Koncept podatkovnega skladišča je prvi predstavil W.H.Inmon v začetku 90ih let. Podatkovno skladišče je definirano kot »vsebinsko usmerjeno (subject oriented), integrirano (integrated), obstojno (non volatile), časovno variantno (time variant) zbirko podatkov za podporo poslovnim odločitvam.« Tako lahko trdimo, je Inmonova definicija najpomembnejša, saj poudarja poglobljene značilnosti podatkovnih skladišč, ter da jo v svojih literaturah uporablja veliko avtorjev.

Podatkovna skladišča vključujejo implementacijo pristopa do heterogenih virov podatkov, prečiščevanje, filtriranje, transformacijo podatkov ter shranjevanje le teh v primerno strukturo, ki je lahko dostopna, razumljiva in uporabna. Kot taka se popolnoma razlikuje od transakcijsko orientiranih operacijskih okolij, saj vsebuje drugačen pristop, razlikuje pa se tudi glede uporabe, tehnologij ter performančnih zahtev. Količina podatkov v podatkovnem skladišču je lahko zelo velika, še posebno takrat, ko želimo analizirati tudi zgodovinske podatke. Kadar v transakcijskem sistemu analiziramo velike količine podatkov, se lahko srečamo z velikimi performančnimi problemi, kar ni ravno zaželeno. Zato je toliko bolj pomembno, da sta omenjena okolja ločena, da se izognemo konfliktom ter degradaciji performanc transakcijskega oziroma operacijskega okolja (Redbook, 1998). Več glede razlik med omenjenima okoljema bo opisano v enem izmed naslednjih poglavij.

Podatkovno skladišče je enotna zbirka podatkov, ki zajema najpomembnejše podatkovne entitete področja organizacije, celotne organizacije ali več organizacij. Iz stališča arhitekture predstavlja temelj sistemov za podporo odločanju. Med najpomembnejše značilnosti podatkovnih skladišč štejemo:

- področna usmerjenost: podatkovno skladišče hrani podatke o najpomembnejših področjih poslovanja, ki so skupni celotni organizaciji;
- statičnost podatkov: podatki v podatkovnem skladišču so namenjeni poizvedovanju. Podatkovno skladišče ni operativna baza!;
- zgodovina: v podatkovnem skladišču hranimo podatke za daljšo zgodovino;
- Integriranost podatkov; podatkovno skladišče se polni iz različnih virov (Bajec, 2005, str. 3).

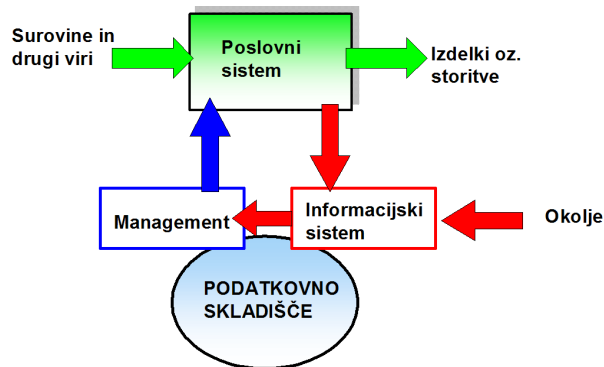
V strokovnih literaturah so s pojmom podatkovno skladišče povezani izrazi kot so sprotna analitična obdelava, poslovno obveščanje, sistemi za podporo procesom odločanja, OLAP, itd. Že iz navedenih pojmov lahko smatramo, da se podatkovno skladišče nekoliko razlikuje od transakcijske baze. kjer se praviloma nahajajo poslovni podatki.

Poslovni podatki so praviloma shranjeni v transakcijskih bazah. Ker se v podatkovni bazi beležijo vse transakcije, je ta baza zelo velika, zaradi velikega števila transakcijskih mest pa tudi polno obremenjena. Transakcijska baza praviloma obsega le podatke omejenega časovnega obdobja, ponavadi enega poslovnega leta. Podatkovno skladišče obsega podatke daljšega časovnega obdobja in agregate podatkov, ki se ne nahajajo v transakcijski bazi (LANCom, 2008).

Podatki niso shranjeni v relacijski bazi, temveč v zato prilagojenih dimenzijskih tabelah.

Prof. dr. V. Rajkovič (1998, str 2.) v svojih člankih omenja, da naj bi imelo podatkovno skladišče vlogo in mesto v procesu upravljanja poslovnega sistema (Slika 1). Namen smiselnega upravljanja poslovnega sistema je doseganje njegovih ciljev, kot so finančna uspešnost, kakovosti proizvodov in storitev, izpolnjevanje rokov ter fleksibilnost v smislu prilagajanja poslovnega sistema novo nastalim situacijam. Tako kompleksno upravljanje, ki zahteva številne odgovorne odločitve lahko izvaja le človek. Pri tem pa si lahko bistveno pomaga z ustreznimi informacijami, ki mu jih nudi informacijski sistem. Ta pa zajema podatke tako iz svojega poslovnega sistema, kakor tudi iz širšega okolja. Podatkovno skladišče lahko razumemo kot vez med informacijskim sistemom in uporabniki – upravljalci. Podatke, ki jih zagotavlja IS približa miselnim procesom človeka, ki jih laže pretvori v potrebne informacije za upravljalško odločanje.

Slika 1: Mesto in vloga podatkovnega skladišča v informacijsko upravljalskem procesu poslovnega sistema



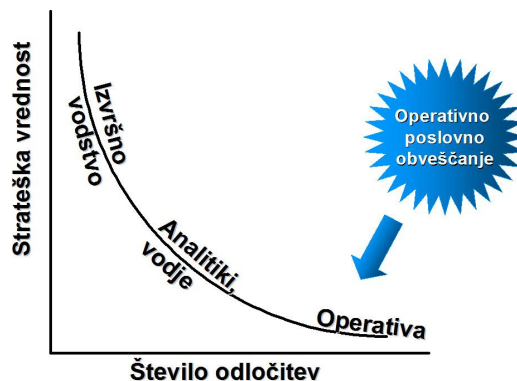
Vir; V. Rajkovič, *Je podatkovno skladišče del splošne informacijske kulture?*, 1989, str. 1

#### 4.1 Poslovno obveščanje

Z izrazom poslovno obveščanje razumemo vse sisteme, ki omogočajo uporabnikom analizo podatkov z namenom razumevanja delovanja organizacije in posledic sprotih odločitev. V večini primerov gre za elemente sistemov za podporo managementu na različnih nivojih. Razvoj poslovne inteligence je tesno povezan z informacijsko demokratizacijo, ki omogoča čedalje večjemu številu uporabnikov možnost dostopa do podatkov in njihovo analizo. Po drugi strani pa tudi pri strateških odločitvah vse pogosteje ne zadošča le presoja na podlagi izkušenj, temveč je za doseganje kratkoročnih prednosti nujna analiza velikih količin podatkov, kar je preprosto mogoče z razvojem zmogljive strojne in programske opreme, nastankom sodobnih integriranih podatkovnih virov, npr. podatkovnih skladišč in navsezadnje z dovolj velikimi količinami zbranih podatkov v digitalni obliki (Jaklič, 2002, str. 178).

Poslovno obveščanje mora torej uporabnikom omogočiti hiter dostop do poslovnih informacij. To je običajno najhitreje preko podatkovnega skladišča. To pomeni manj časa za zbiranje podatkov in več časa za analizo podatkov in ukrepanje. Razmerje med strateško vrednostjo in številom odločitev dobro prikazuje spodnja slika.

Slika 2: Nov potencial poslovnega obveščanja



Vir: J. Berce, *Analysis Services v MS SQL Server 2005, 2004*, str. 3

V okviru informacijske tehnologije za poslovno obveščanje poznamo danes od orodij za poizvedovanje po podatkovnih virih, preko orodij za sprotno analitično obdelavo podatkov, do orodij za rudarjenje po podatkih in ostalih specialnih orodjih za analizo.

Ker pa zaradi velike količine podatkov uporabnikom zahtevna, težko razumljiva poročila niso bila v pomoč, je kasneje sprotna analitična obdelava podatkov ali OLAP (angl. *on-line Analytical Processing*) omogočala pridobiti prave podatke ob pravem času.

Že v uvodu je bilo omenjeno, da je koncept podatkovnega skladišča prvi predstavil W.H.Inmon v začetku 90ih let, ki je podatkovno skladišče definiral kot »vsebinsko usmerjeno (angl. *subject oriented*), integrirano (angl. *integrated*), obstojno (angl. *non volatile*), časovno variantno (angl. *time variant*) zbirko podatkov za podporo poslovnim odločitvam«. Tako lahko trdimo, je Inmova definicija najpomembnejša, saj poudarja pglavitne značilnosti podatkovnih skladišč, ter da jo v svojih literaturah uporablja veliko avtorjev.

Kelly (1994) opisuje podatke v podatkovnem skladišču, ki so:

- ločeni od ostalih transakcijskih sistemov v podjetju in sestavljeni iz podatkov teh sistemov. Ločenost od podatkov v transakcijskih sistemih je pomembna, ker s tem zagotovimo nemoteno obdelavo transakcij tudi ob zelo zahtevnih poizvedbah, ki bi sicer lahko upočasnile izvajanje transakcijskih aplikacij;
- namenjeni izključno za povpraševanja poslovnih uporabnikov. Namen tako zbranih podatkov je torej podpora poslovnemu odločanju;
- dostopni uporabnikom, ki imajo omejeno znanje o računalniških sistemih ali podatkovnih strukturah. Ni torej dovolj, da so podatki dostopni informatikom in zahtevnejšim uporabnikom z znanjem strukture podatkovnih zbirk in poizvedovalnih jezikov (npr. sql,...). Podatki morajo biti dostopni vsem, ki jih potrebujejo za pomoč pri poslovnih odločitvah, torej tudi analitikom ter srednjemu in najvišjemu sloju upravljalcev.

Skladišče podatkov torej omogoča, da se podatki različnih sistemov združijo v enotno zbirko podatkov, se obravnavajo na enoten način in so dostopni vsem pooblaščenim uporabnikom. Hranijo se lahko tudi zgodovinski podatki zaradi pregleda po časovnih vrstah, česar transakcijski sistemi običajno ne omogočajo. V skladišče podatkov je mogoče prenesti razen podatkov in poslovnih informacijskih sistemov podjetja tudi podatke, pridobljene iz zunanjih virov. Proces prenosa podatkov v skladišče imenujemo s tujo kratico ETL – (angl. *Extract Transform Load*). ETL proces pa še daleč ni le enostaven proces polnjenja dimenzijskih tabel, temveč kot pravi Ralph Kimball, so etl procesi v življenjskem ciklu izgradnje podatkovnega skladišča daleč najtežji, časovno zahtevni in delovno intenzivni (Kimball, 2004).

Iz zgoraj navedenega lahko povzamemo, da je namen podatkovnih skladišč zagotoviti dostopne, dosledne in skladne informacije. Podatkovno skladišče mora biti prilagodljiv in prožen vir informacij, ki morajo biti varovane in zaščitene ter nam služiti za podporo procesom odločanja. Ali kot pravi oče podatkovnih skladišč R. Kimball (1998), da so se podatkovna skladišča razvila zato, ker iz ogromne količine podatkov, ki se zbirajo v transakcijskih sistemih ni mogoče na preprost način dobiti informacije o poslovanju za potrebe analiz in odločanja. S pomočjo transakcijskih sistemov se podatki zbirajo, s pomočjo podatkovnih skladišč pa se oblikujejo podatki za hitro pripravo izhodnih informacij o poslovanju. Ti dve obliki ali sistema se razlikujeta po tem, kdo so uporabniki, kakšne so



uporabniške zahteve in kakšne so časovne zahteve. Več na to temo je opisano v poglavju lastnosti in razlike med OLTP in OLAP.

## 4.2 OLAP – sprotna analitična obdelava podatkov

Že zgoraj je bilo omenjeno, da so to orodja za sprotno analitično obdelavo podatkov. Kompleksne analize podatkov so oteževale hitro odločanje, kar je v današnjem času bistvenega pomena. In rešitev so bila prav orodja OLAP, ki omogočajo neposreden dostop do podatkovnih virov in izdelavo poljubnih pogledov na podatke. Omenjena orodja nam omogočajo, da poslovni dokumenti postanejo živi, interaktivni in s tem mnogo bolj transparentni, kar nam omogoča boljše razumevanje podatkov in s tem pridobivanje informacij za boljše odločanje.

Managerjem je omogočeno, da si sami na enostaven način pripravijo pogled na podatke, ki ga potrebujejo za določeno situacijo ter da z enostavnim spreminjanjem določenega pogleda lahko ugotavljajo, kateri podatki so zanimivi in relevantni za sprejemanje poslovnih odločitev (Jaklič, 2002, str 178., Kavčič, 2005).

OLAP torej uporabnikom omogoča na izredno fleksibilen in samostojen način dostopati do podatkov. Predpogoj za učinkovito pridobivanje informacij iz podatkov pa je predhodno dobro pripravljen vir podatkov v podatkovni bazi in uporabniku enostavna in razumljiva uporaba orodja.

Termin OLAP je leta 1993 uvedel E. F. Codd, ki je znan kot oče relacijskih baz podatkov (Gray & Watson, 1999). OLAP je kratica za Analytical On Line Processing oziroma po slovensko sprotno analitično procesiranje (Marolt-Šmid, 2001). Gre za tehniko pregledovanja podatkov in izdelave analiz po več dimenzijah (večdimenzionalno opazovanje). Za analize se uporabljajo agregirani in sumirani podatki. OLAP omogoča končnemu uporabniku v realnem času pridobiti strateške informacije iz podatkovnih skladišč ali operativne baze podatkov (ad-hoc poizvedbe). S klikom miške lahko uporabnik dobi poljubne prereze podatkov in aggregate.

OLAP tehnika omogoča (Watson, 1999):

- večdimenzionalen pregled podatkov;
- preglednost;
- dostopnost podatkov;
- več-uporabniško podporo;
- operacije med različnimi nivoji;
- poročanje;
- arhitekturo odjemalec – strežnik.

## 4.3 Podatkovno skladišče in področno podatkovno skladišče

V tujih in domačih strokovnih literaturah lahko zasledimo dva pojma, in sicer podatkovno skladišče in področno podatkovno skladišče. Zadnjega nekateri poimenujejo tudi podatkovne tržnice (angl. *data mart*). Zgodnje teorije so razlagale, da področna podatkovna skladišča sestavljajo podatkovno skladišče, vendar se je v praksi izkazalo, da ni ravno tako. Kimball (1997) trdi, da so področna podatkovna skladišča logična podmnožica celotnega

podatkovnega skladišča. Podatkovno skladišče opisujejo tabele dimenzij in dejstev, zato morajo imeti področna podatkovna skladišča usklajene dimenzijske tabele in usklajene tabele dejstev. Podatkovno skladišče je unija vseh njegovih področnih podatkovnih skladišč. Zadaž za to preprosto definicijo se večkrat skriva pogled na področno podatkovno skladišče, ki predstavlja omejeno podatkovno skladišče. Obravnava le en poslovni proces ali grupo odvisnih poslovnih procesov, ki ciljajo na eno določeno poslovno skupino. Podatkovno skladišče pa pomeni tako organizirane podatke organizacije, da omogočajo vsakovrstno poizvedovanje.

Področno podatkovno skladišče je podatkovni vir prirejen za uporabo v sistemih za podporo odločanju za posamezna poslovna področja (finance, prodaja, trženje, itd.) (Jaklič, 2002, str. 20).

POE (1996, str 40-49) opisuje različice podatkovnih skladišč oziroma prepletanja med centralnim področnim skladiščem na sledeči način. Končna rešitev podatkovnega skladišča je v vsaki organizaciji nekoliko drugačna, saj je že začetno stanje pred gradnjo skladišča podatkov v vsaki organizaciji posebnost zase. Organizacija ima lahko eno centralno skladišče podatkov, iz katerega se napaja več manjših skladišč podatkov, izmed katerih vsako vsebuje podatke za določeno poslovno področje in potrebuje svojo skupino analitikov za analiziranje. Podskupina skladišča podatkov, ki se usmerja na eno ali več ožjih poslovnih področij, se imenuje področno podatkovno skladišče (angl. *Data Mart*), ki se lahko polni tudi direktno iz virov podatkov in ne posredno, preko skupnega skladišča podatkov, kot v prejšnjem primeru. Področno podatkovno skladišče je optimizirano za pregledovanje in analiziranje, vsebuje visoko agregirane ter zgodovinske podatke in je največkrat shranjena kar pri odjemalcu. Uporabljajo jo lahko, na primer, potniki na svojem prenosnem računalniku.

Ne glede na to ali je podatkovno skladišče majhno ali veliko ali imamo področna podatkovna skladišča ali le eno centralno skladišče, je organizacija le tega odvisna od arhitekture podatkovnega skladišča, ki določenemu podjetju odgovarja. Bistveno je, da podatkovno skladišče služi svojemu namenu ter nam pravi in razumljivi kazalci dajejo upravljalno pomembne informacije.

#### **4.4 Arhitekture podatkovnih skladišč**

V začetku načrtovanja podatkovnega skladišča je ena najpomembnejših odločitev izbira primerne arhitekture, saj le-ta določa podatkovni model, vlogo področnih skladišč ter sosledje korakov v razvojnem ciklu. Vendar je prav na področju poznavanja in razumevanja arhitektur podatkovnih skladišč največ zmede. K temu dodatno prispevajo še avtorji in zagovorniki posameznih arhitektur, ki opozarjajo na neprimernosti ostalih arhitektur.

V nadaljevanju bodo predstavljene tri arhitekture podatkovnih skladišč: centralizirano, distribuirano in federativno.

Centralizirana arhitektura podatkovnega skladišča je sestavljena iz osrednjega podatkovnega skladišča in iz množice področnih skladišč. Značilnost te arhitekture je, da se področna skladišča polnijo izključno iz osrednjega. Osnovni uporabljen podatkovni model je E-R model, uporabljena metodologija izgradnje sistema pa ne priporoča izpeljave intervjujev kot načina za pridobitev zahtev uporabnikov o podatkovnem skladišču. Temeljni razlog za to so preveč spremenljive zahteve uporabnikov.

Distribuirana arhitektura temelji na množici povezanih, a samostojnih področnih shramb, ki tvorijo podatkovno skladišče. Pomemben koncept je skupno podatkovno vodilo, ki omogoča izgradnjo podatkovnega skladišča. Osnovni uporabljen model je dimenzijski, predlagana metodologija pa priporoča izpeljavo intervjujev že na samem začetku.

Federativna arhitektura je novejša, hibridna rešitev, ki temelji na skupnem poslovnem modelu. Trenutno je njena največja pomanjkljivost, da je še v fazi uveljavitve in da še ni sprejeta s strani proizvajalcev sistemov za podatkovno skladiščenje.

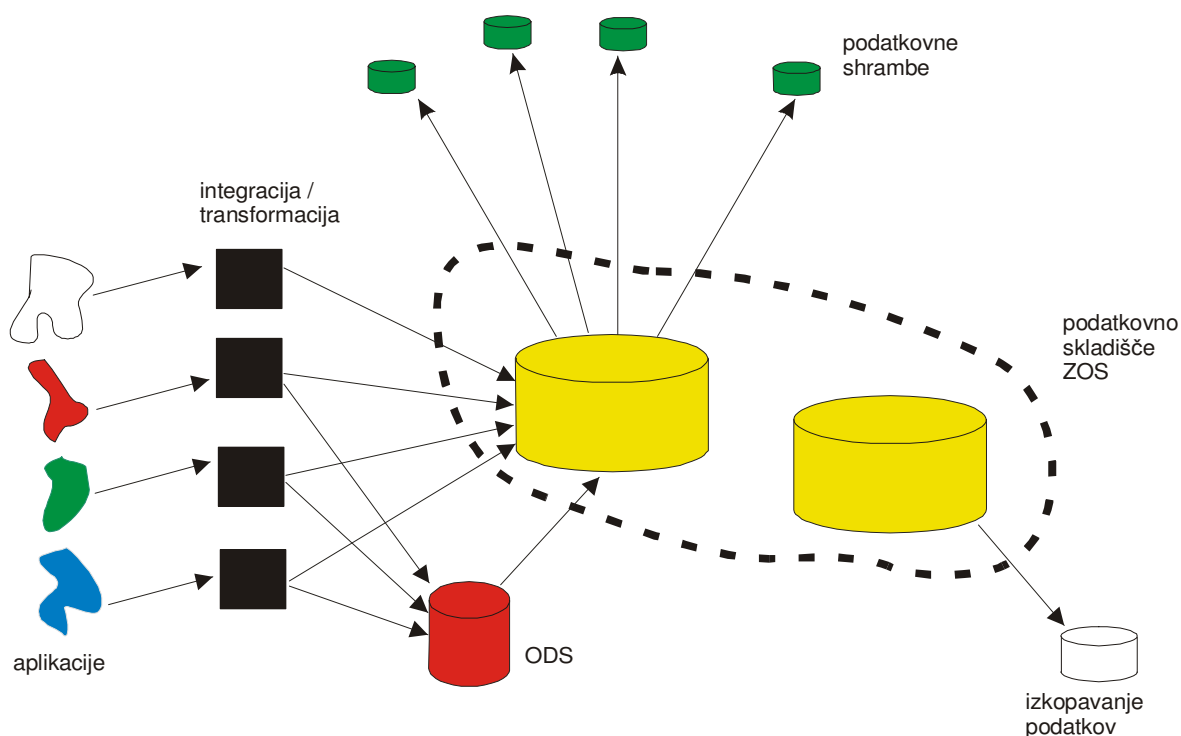
Za določitev učinkovite in ustrezne arhitekture podatkovnega skladišča, ki rezultira v zagotavljenem primernem odzivnem času in učinkoviti izrabi virov, je nujno poznavanje lastnosti posameznih arhitektur, pozitivnih in negativnih. Pravilna odločitev zmanjša tveganje v novih projektih podatkovnega skladiščenja (po nekaterih virih propade kar 70% projektov podatkovnih skladišč), prispeva k izboljšanju obstoječih podatkovnih skladišč z novimi področnimi skladišči zaradi odprtosti sistemov ter omogoča prilagoditev preveč centraliziranih podatkovnih skladišč, ki ne morejo učinkovito delovati v nehierarhičnem okolju, na bolj demokratično arhitekturo (Golob, Welzer, 2000, str 8-9).

#### **4.4.1 Centralizirana arhitektura**

V središču centralizirane arhitekture podatkovnega skladišča je podatkovno skladišče zaključenega organiziranega sistema, ki »hrani« področna skladišča, polni pa se iz operativnih podatkovnih baz ter operativnega podatkovnega skladišča. Največji zagovornik takšne arhitekture je Inmon (Inmon, 1996). Strogo rečeno so v taki arhitekturi področna skladišča odvisna struktura, saj so podatki pridobljeni oz. naloženi izključno iz podatkovnega skladišča organizacije.

Osnovno arhitekturo centraliziranega podatkovnega skladišča prikazuje slika 3:

Slika 3: Arhitektura centraliziranega podatkovnega skladišča



Vir: T. Welzer & I. Golob, *Arhitekture podatkovnih skladišč*, 2001, str. 3.

Razlike med operativnim svetom in svetom podatkovnega skladišča so izrazito vidne tudi iz opisa razvojnih ciklov. Operativno okolje je podprto s klasičnim razvojnim ciklom, ki je voden s strani zahtev. Tako je potrebno najprej razumeti zahteve, šele nato preidemo v faze načrtovanja in razvoja. Razvojni cikel podatkovnega skladišča je podatkovno voden, saj pričenmo s podatki. Po integraciji podatkov pogledamo, če je potrebno njihovo dodatno uglaševanje in to po potrebi tudi storimo. Rezultati programov so analizirani, in šele na koncu razumemo zahteve. Vidimo, da je vrstni red posameznih faz razvojnega cikla operativnega okolja in okolja podatkovnega skladišča tako popolnoma obrnjen (Welzer, 2000).

#### 4.4.2 Distribuirana arhitektura podatkovnega skladišča

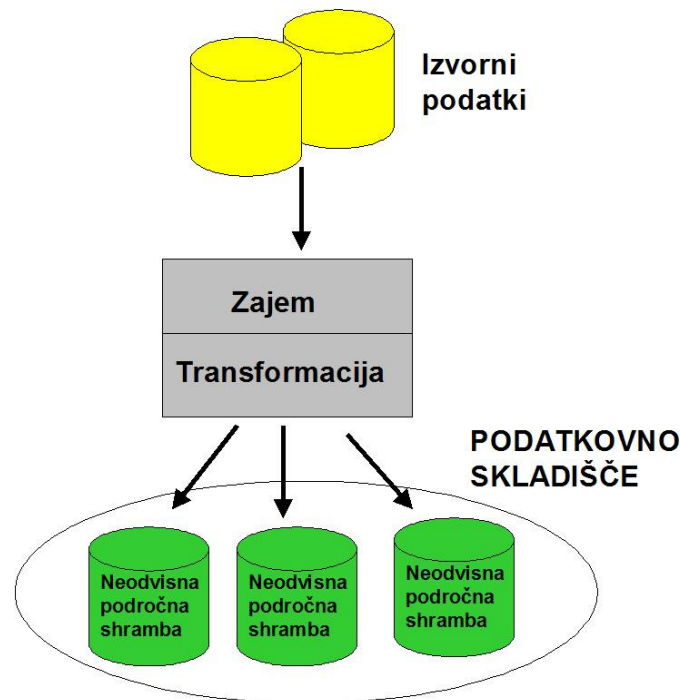
Področno skladišče je podmnožica podatkovnega skladišča določene organizacije. V distribuirani arhitekturi je podatkovno skladišče le unija področnih skladišč. Področno skladišče igra ponavadi vlogo oddelčnega, krajevnega ali funkcionalnega podatkovnega skladišča in podpira eno ali več specifičnih področij.

Tipično distribuirano arhitekturo, katere največji zagovornik je Kimball, prikazuje slika 4.

Organizacija kot del iterativnega procesa gradnje podatkovnega skladišča zgradi vrsto distribuiranih področnih skladišč in jih na koncu poveže v logično podatkovno skladišče celotne organizacije. Hackney tak pristop brez zadržkov poimenuje "od-zgoraj-navzdol" (angl. *bottom-up*) (Hackney, 1997 & Golob, Welzer, 2000).

Področna skladišča postavljajo specifične oblikovalske zahteve. Vsako področno skladišče mora biti predstavljeno z dimenzijskim modelom, ki mora biti znotraj enotnega podatkovnega skladišča skladen.

Slika4: Distribuirano podatkovno skladišče



Vir: T. Welzer & I. Golob, *Arhitekture podatkovnih skladišč*, 2001, str. 5.

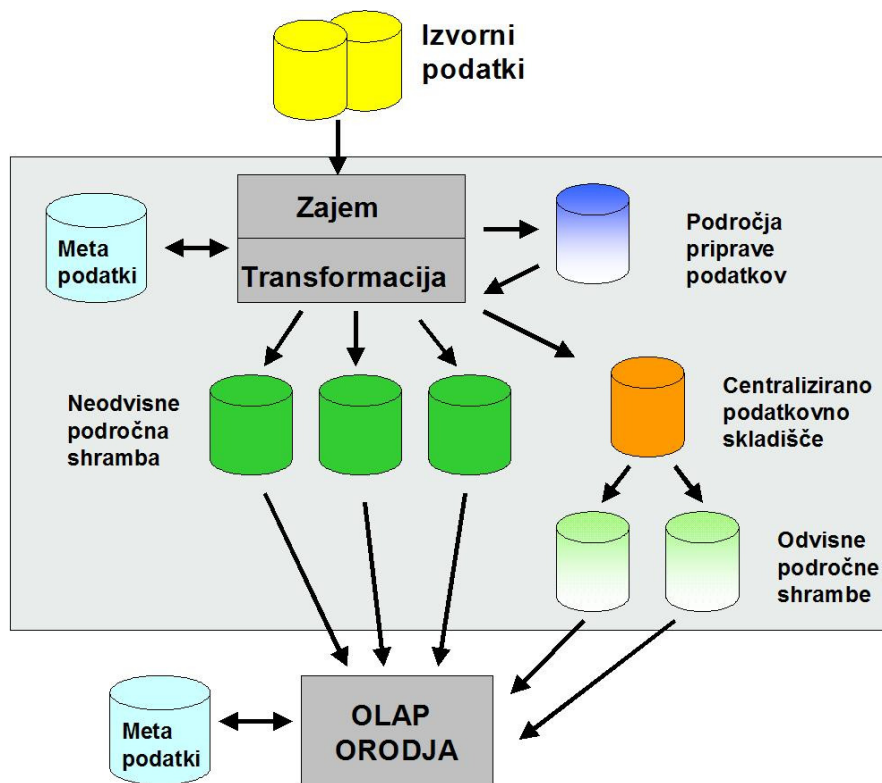
Skladna dimenzija (angl. *conformed dimension*) je dimenzija, za katero je značilno, da ima enoličen pomen, ne glede na to, s katero tabelo dejstev jo povežemo. Zagotavlja tudi, da je podatek predstavljen le enkrat. Glavna naloga skupine načrtovalcev podatkovnega skladišča pri načrtovanju distribuirane arhitekture podatkovnega skladišča je vzpostavitev, objava in vzdrževanje skladnih dimenzij, kot tudi zagotavljanje njihove dosledne uporabe. Brez upoštevanja koncepta skladnih dimenzij podatkovno skladišče ne more delovati kot integrirana celota.

#### 4.4.3 Federativna arhitektura podatkovnega skladišča

Federativno podatkovno skladišče (angl. *federated data warehouse*) je hibridna rešitev, temelječa na skupnem poslovnem modelu (angl. *common business model*) in področjih priprave informacij (angl. *information staging areas*), ki so v skupni rabi. Predlagana arhitektura zagotavlja nizke stroške in hitro povrnitev vloženi sredstev z uporabo neodvisnih področnih skladišč, pri čemer kasnejša podatkovna integracija ni potrebna.

Slika 5 prikazuje arhitekturo federativnega podatkovnega skladišča:

Slika 5: Federativna arhitektura podatkovnega skladišča



Vir: T. Welzer & I. Golob, *Arhitekture podatkovnih skladišč*, 2001, str. 6.

#### 4.4.4 Skupni informacijski (poslovni) model

Opisan pristop podpira iterativni razvoj podatkovnega skladišča, ki vsebuje neodvisna področna skladišča. Ključni element podatkovne integracije je skupni poslovni model poslovnih informacij, hranjen in upravljan s strani podatkovnega skladišča. Izdelava skupnega poslovnega modela zagotavlja konsistenco pri uporabi imen podatkov in poslovnih definicij v vseh procesih podatkovnega skladišča.

Skupni poslovni model se ažurira vsakič, ko se zgradi novo področno skladišče. Kadar obstoječi podatki v operativnih sistemih narekujejo načrtovanje področnih skladišč, uporabimo skupni poslovni model (in ga po potrebi ažuriramo), vzporedno z razvojem podatkovnih modelov podrejenih področnih skladišč. V okoljih, kjer so vodilo poslovne zahteve uporabnikov iz poslovnega sveta, najprej razvijemo skupni poslovni model in ga nato uporabimo za razvoj podatkovnih modelov podrejenih področnih skladišč. Tak razvoj podpira tudi več obstoječih rešitev. Pri uporabi analitične programske opreme "iz škatle" (kot je npr. tipično orodje OLAP) pa je pomembno, da je možna sprememba poslovnih modelov in njihova integracija v skupni poslovni model organizacije. V takem primeru razvoj skupnega poslovnega modela in pripadajočih podatkovnih modelov močno pospešijo morebitne predloge poslovnih področij (angl. *business area templates*). Ta dokumentirajo poslovne metrike in pravila, uporabljena pri analizi in modeliranju poslovnega procesa.

## Področja priprave informacij

Vsakič, ko je zgrajeno novo področno skladišče, se praviloma ustvari nova garnitura aplikacij za zajem in transformacijo, ki pa so le redko integrirana z aplikacijami za gradnjo ostalih področnih skladišč. In prav tu je past: uporaba neodvisnih področnih skladišč poveča namreč tudi število programskih rutin za zajem in transformacijo.

Vzdrževanje takih rutin je izjemno kompleksno in zahtevno do virov, najtežje pa je zagotoviti njihovo izvajanje v času, ko je operativna baza obremenjena z rednim procesom. Rešitev omenjenega problema je razbitje procesiranja v več korakov, kar vključuje razvoj množice rutin, ki zajemajo podatke in z njimi polnijo področja priprave. Podrobni podatki iz teh področij pa se nato pretočijo v neodvisna področna skladišča s pomočjo orodij ETL (orodij za zajem, transformacijo in polnjenje, angl. *Extract-Transform-Load tools*) v obliko, ki jo zahteva ciljna področno skladišče.

Ob dodajanju novih področnih skladišč v sistem podatkovnega skladišča se obstoječe rutine za zajem in podatki v področjih priprave po potrebi ponovno uporabijo oziroma izboljšajo. Tak način še posebej dobro deluje v federativnem podatkovnem skladišču, kjer lahko skupni poslovni model uporabimo pri načrtovanju področij priprave in programskih rutin za zajem podatkov.

Uporaba skupnih področij uporabe lahko prav tako pripomore pri izboljšanju kakovosti zajetih podatkov, saj lahko v njih že pred uporabo orodij ETL (ki večkrat niso sposobna obvladovati široke množice problemov) uporabimo posebna orodja za prečiščevanje. Tako za polnjenje različnih struktur uporabimo prečiščene, konsistentne podatke iz področij priprave. Uporaba orodij za prečiščevanje in orodij za povratni inženiring pred uporabo orodij ETL zmanjšuje probleme s kakovostjo podatkov v podatkovnem skladišču in lahko v veliki meri pripomore k večjemu uspehu projekta podatkovnega skladišča in k hitrejši povrnitvi vloženih sredstev. Tak pristop imenujemo tudi analizo voden (angl. *analysis-led*) in je nasproten ETL-vodenemu (angl. *ETL-driven*). Kljub vsemu pa se je potrebno zavedati, da analizo voden pristop - kljub izboljšavam, ki jih prinaša - ni "panacea" za problem kakovosti podatkov (Golob & Welzer, 2000).

## 4.5 Lastnosti in razlike med OLTP in OLAP

Konkurenčna prednost in učinkovitost podjetij temelji na pravočasnih in točnih informacijah. Velikokrat lahko že iz operativnih baz pridobimo potrebne informacije, vendar naj bi v praksi veljalo, da se v operativnih bazah zbirajo podatki o dogodkih, medtem ko naj bi iz podatkovnih skladišč, kamor se ti podatki prenesejo, pridobili informacije, ki naj bi omogočale lažje poslovno odločanje.

Na operativnem in analitičnem nivoju se potrebe po podatkih močno razlikujejo. Na operativnem nivoju je za uporabo podatkov značilno:

- podatki se ne bi smeli podvajati, da se izognemo težavam pri njihovem posodabljanju,
- za uporabnike so zanimive predvsem trenutne vrednosti podatkov,
- količina podatkov, s katerimi se dela, je ponavadi majhna, dostopanje do podatkov pa pogosto,
- redko se spreminja način uporabe,
- čas dostopa mora biti kratek, zanesljivost podatkov pa visoka.

Za uporabo podatkov na analitičnem nivoju pa velja:

- podvajanje ne predstavlja problema, ker se podatke večinoma ne posodablja,
- za uporabnike so zanimivi tudi zgodovinski podatki,
- količina podatkov, s katerimi se dela, je ponavadi velika, vendar analitik dobi kot rezultat le nekaj sumarnih vrednosti, dostopanje do podatkov pa je neenakomerno,
- podatki se uporabljajo zelo raznoliko,
- za zahtevnejše analize so uporabniki pripravljeni počakati nekaj sekund ali minut, morda celo ur, zanesljivost podatkov pa ni tako pomembna kot na operativnem nivoju (Jaklič, 2002, str. 16-17).

Najbolje prikazuje razliko med podatki v OLAP in OLTP sistemu naslednja preglednica.

*Tabela 1: Razlike med OLTP in OLAP*

	<b>OLTP</b> (angl. <i>On Line Transaction Processing</i> ) <b>Sprotno transakcijsko procesiranje</b>	<b>OLAP</b> (angl. <i>On Line Analytical Processing</i> ) <b>Sprotno analitično procesiranje</b>
1.	Podatke se v glavnem vnaša, spreminja in briše. Vsaka operacija nad podatki se zapiše v bazo podatkov, ki je najpogosteje relacijska	Podatke se v glavnem bere. Lahko se jih tudi spreminja, dodaja in briše za »kaj-če« analize, vendar se te spremembe ne vpisujejo v bazo
2.	Veliko število dogodkov	Malo število dogodkov
3.	Podatki se zaradi pogostih vnosov nenehno spreminjajo, osvežujejo	Podatki so statični. Osvežijo se le po vnaprej definiranem urniku in ne takrat, ko pride do sprememb v operativni bazi
4.	Veliko uporabnikov, ki delajo z majhno količino podatkov znotraj ene transakcije	Malo uporabnikov, ki delajo z veliko količino podatkov znotraj ene analize
5.	Uporabne za transakcijsko obdelovanje	Uporabne za analitične poizvedbe
7.	Časovna enota podatka so dnevi	Ad-hoc poizvedbe uporabnikov
8.	Podatkovni model je zelo normaliziran, z velikim številom tabel in povezav	Podatkovni model je zaradi boljše odzivnosti namerno denormaliziran
9.	Sistem mora biti vedno dostopen	Občasna nedostopnost sistema ni kritična
10.	Zahteva se zelo hitra odzivnost sistema	Odzivnost sistema ni tako pomembna
11.	Utečen informacijski sistem je dokončan	Skladišče podatkov ni nikoli dokončano
12.	Podatki so tekoči	Podatki so tekoči in predvsem zgodovinski
13.	Atomski podatki	Atomski in agregirani podatki
14.	Majhno število atomskih podatkov	Veliko število atomskih podatkov
15.	Po obsegu niso velike	Po obsegu so lahko zelo velike
16.	Podatki so organizirani po aplikacijah	Podatki so organizirani glede na interes zanimanja, glede na dimenzije poslovanja
17.	Odvečnosti podatkov je odpravljena z uporabo referenčne integritete	Podatki se mnogokrat ponavljajo, predvsem zaradi boljših odzivnih časov poizvedb
18.	Uporabnika na zanima struktura podatkov	Struktura podatkov mora biti bralno razumljiva uporabnikom

»se nadaljuje«



»nadaljevanje«

	<b>OLTP</b> (angl. <i>On Line Transaction Processing</i> ) <b>Sprotno transakcijsko procesiranje</b>	<b>OLAP</b> (angl. <i>On Line Analytical Processing</i> ) <b>Sprotno analitično procesiranje</b>
19.	Ne omogočajo shranjevanja zgodovinskih podatkov, ker operativno niso zanimivi	Lahko shranjujejo ogromne količine zgodovinskih podatkov
20.	Podatkov med sabo ni mogoče primerjati	Zaradi statičnih in zgodovinskih podatkov je mogoča njihova medsebojna primerjava
21.	Ne omogočajo agregiranja podatkov	Agregirani podatki so glavna značilnost
22.	Podatki so neintegrirani	Podatki so visoko integrirani
23.	Primarni cilj je zagotavljanje operativnega poslovanja podjetja	Primarni cilj je analiziranje trendov, problemov ali določenih situacij

Vir: B. Kos, *Izgradnja skladišča podatkov*, 1997, str 28, Tabela 4.

#### 4.6 Razlogi za neuspeh podatkovnih skladišč

V praksi pa niso vse uvedbe podatkovnih skladišč uspešne. Razlogi za neuspeh so lahko različni:

- Zavedati se moramo, da gradnja podatkovnega skladišča ni nikoli dokončna. Potrebna so nenehna dograjevanja in prilagajanja uporabnikom.
- Z večanjem količine podatkov lahko posvečamo premalo pozornosti odzivnosti podatkovnega skladišča.
- Premajhna podpora vodstva za uvedbo podatkovnega skladišča. Premalo podjetij se zaveda, da je ključnega pomena v začetni fazi dobra predstavitev po potrebi podatkovnega skladišča vodilnim v podjetju, saj si s tem zagotovimo potenciranje potreb po uvedbi in uporabi tudi na nižje nivoje organizacije.
- Prevelike obljube med predstavitvijo potreb po gradnji skladišč podatkov povzročijo poznejša razočaranja, ko rezultati ne dosegajo obljub. Pozorni moramo biti tudi pri predstavitvi posameznim vodjem in analitikom, da ne bodo dobili občutka, da do sedaj niso dovolj dobro delali. Potrebno jih je motivirati na način, da bodo z novo tehnologijo lahko izvajali analize, ki do sedaj niso bile omogočene ter da jim bo s tem olajšana priprava podatkov ter lažje odločanje.
- Uvedba preobsežnega skladišča podatkov ni smiselna, saj se lahko izgubimo v podrobnostih ter oddaljimo od začetnega cilja, kateri podatki bodo zares potrebni oziroma kaj naj bi podatkovno skladišče omogočalo.
- Premajhna pozornost pri začetnem zbiranju zahtev ter definiranju cilja podatkovnega skladišča.
- Nenatančni in nepopolni opisi podatkov lahko v podatkovnem skladišču povzročijo nepravilno razumevanje pomena podatkov pri uporabnikih in s tem nevarnost nepravilne uporabe le teh.
- Da se lahko zadosti obljubam po zmogljivosti, je včasih potrebno dokupiti dodaten medij za podatke ali dodaten računalniški procesor, da lahko dosežemo zelene zmogljivosti. Najobčutljivejša točka, ki vpliva na odzivni čas sistema je, propustnost računalniškega omrežja. Podjetja se včasih ne zavedajo, da gradnja skladišča podatkov zahteva veliko začetno naložbo, saj je praviloma potrebno nakupiti povsem novo strojno, programsko in komunikacijsko opremo ter plačati delo razvojne skupine.

- Zmotno je mišljenje, da je uvajanje podatkovnega skladišča preprosto. Ni dovolj imeti le dobrega analitičnega poizvedovalnega orodja, že samo polnjenje, pridobivanje ter transformacija podatkov zahteva posebno pozornost.
- Misliti, da so možne samo te napake je napačno. Gradnja skladišča podatkov je zahtevno in odgovorno opravilo, zato je logično pričakovati, da bomo v začetku ali kasneje lahko naleteli na težave.
- Razlogi za neuspeh je lahko tudi pomanjkanje pozitivnega odnosa do informacijske podpore. Uporabniki lahko niso navdušeni nad pridobivanjem informacij iz podatkovnega skladišča, saj imajo premalo znanja za definiranje informacijskih zahtev ter uporabe orodja. Tako tudi iz razpoložljivih prikazov ne znajo povzeti uporabnih informacij.
- Uporabniki imajo lahko težave pri uporabi orodja za analiziranje podatkov, saj lahko niso tako jasna in preprosta za uporabo kot trdijo proizvajalci.
- Uporabniki so si predstavljali čisto nekaj drugega, kar so dobili.
- Za neuspeh je lahko vzrok napačno zasnovan in zgrajen podatkovni model baze podatkov skladišča, saj ne ustreza povsem potrebam analiziranju vnaprej izbranih področij poslovanja.
- Če pri gradnji skladišča podatkov delamo napake oziroma ne razrešimo ustrezno težav, se lahko zgodi, da bo potrebno skladišče podatkov nanovo načrtovati in graditi.
- Pri gradnji podatkovnih skladišč se lahko srečamo s težavami kot so: vsi viri podatkov niso na razpolago, niso uporabljeni vsi podatki, uporabljen je lahko samo določen nabor podatkov, za katerega je nekoč nekdo mislil, da je koristen ter podatki, ki zasedejo največ pomnilniškega prostora, morda niso skladiščeni (Berry, 2000),
- Uvedba skladišča podatkov je zahteven, dolgotrajen in drag proces, ker je tudi razlog, da včasih ne uspe (Kos, 1997)

## 5 Podatkovno rudarjenje

V angleških literaturah ga poimenujejo Podatkovno rudarjenje – angl. *Data mining* (DM) ali odkrivanje znanja v podatkih – angl. *Knowledge Discovery in databases* (KDD). Podatkovno rudarjenje se deli na dve veji, in sicer: testiranje hipotez ter odkrivanje znanja.

Podatkovno rudarjenje je predhodno odkrivanje neznanih vzorcev in povezav v podatkih. Je interdisciplinarno področje, ki vključuje baze podatkov, strojno učenje (angl. *machine learning*), zaznavanje vzorcev (angl. *pattern recognition*), statistiko, vizualizacijo in še mnogo drugega. Podatkovno rudarjenje je lahko učinkovito vključeno v mnoga področja, kot so: marketing, prodaja, skladiščno poslovanje, finančno področje, planiranje (SoulEuNet, 2003).

Podatkovno rudarjenje bi bilo lahko učinkovita metoda tudi za področje zdravstvenega zavarovanja. Pestro poslovanje zahteva ogromno analiziranja in odločanja na podlagi zbranih podatkov. Z mojega osebnega vidika bi bilo poleg učinkovite uvedbe in uporabe podatkovnih skladišč potrebno posebno pozornost nameniti tudi podatkovnemu rudarjenju. V teorijah se že kar nekaj časa govori o omenjenem področju, v praksi pa se podatkovnemu rudarjenju posveča premalo pozornosti.

Vir za izkopavanje predstavlja skladišče podatkov. Pomembno je, da je na voljo veliko najrazličnejših podatkov, ker je s tem zmanjšan učinek statističnih napak, pravila poslovanja

pa se lahko skrivajo v na videz še tako nepomembnih podatkih. Za izkopavanje podatkov potrebujemo ustrezne algoritme, ki izvirajo iz statistike, umetne inteligence in učenja strojev. Med algoritme za izkopavanje podatkov sodijo nevronske mreže, odločitvena drevesa, genetski algoritmi, metode najbližjih sosedov in pravila na osnovi statističnih ugotovitev. Izkopane podatke je potrebno predstaviti v obliki grafov ali diagramov, saj imajo številčni podatki majhno informacijsko vrednost.

Izkopavanje podatkov je ena izmed najnovejših tehnologij v zvezi s skladišči podatkov. Izraz izvira iz podobnosti med izkopavanjem podatkov iz velikih baz podatkov in izkopavanjem dragocenih surovin iz rudnikov. V obeh primerih je treba prečesati gore snovi, da bi našli zlato jamo. Tako je iz poslovnega vidika pri izkopavanju podatkov potrebno odkriti dragocene poslovne informacije in skrite povezave med podatki, ki jih na prvi pogled ni mogoče videti. Odkrita poslovna pravila omogočajo, da podatke globlje razumemo in jih tudi koristneje uporabimo. Na osnovi preteklih smeri lahko sklepamo na prihodnje, kar je lahko ena izmed osnov za načrtovanje prihodnjega poslovanja.

## 6 Razvoj podatkovnega skladišča

### 6.1 Načrtovanje podatkovnega skladišča

Načrtovanje je ena izmed najbolj obsežnih faz metodologije razvoja skladišča in obsega:

- **Izbira poslovnega področja za razvoj skladišča podatkov** (POE, 1996).
  - določiti je potrebno, katera področja v organizaciji bodo uporabljala skladišče podatkov;
  - pred gradnjo skladišča podatkov v organizaciji je smiselno izvesti notranji marketing in tako ugotoviti, za katero področje je smiselno graditi skladišče;
  - upoštevati moramo jasen cilj in namen skladišča podatkov. Nekatera poslovna področja so mogoče boljše izbira od drugih za njihovo uresničitev;
  - področja, kot so finance, marketing, prodaja, oddelki za analiziranje, preračunavanje, planiranje in napovedovanje, so navadno najbolj ustrezna za gradnjo skladišča;
  - smiselno je sodelovati s poslovnim področjem, ki je navdušeno nad skladiščem;
- **Definiranje cilja in namena projekta.**  
**Cilj in namen morata biti točno določena in zapisana. Poznati ju morajo vsi člani skupine.**
- **Definiranje lastnosti in prednosti končnega projekta.**
- **Definiranje čim bolj realnega časa gradnje.**
- **Definiranje vseh potrebnih sredstev tako notranjih kot zunanjih.**
- **Oblikovanje skupine za načrtovanje skladišča podatkov.**

Postopek načrtovanja skladišča podatkov je uspešnejši, če ga vodi oseba, ki je po strokovnosti bolj analitik kot računalnikar. V skupino je poleg strokovnega vodja projekta potrebno vključiti še:

- strokovnjake za načrtovanje projektnega načrta in gradnje skladišča podatkov;
- strokovnjake za načrtovanja arhitekture in infrastrukture skladišča podatkov;
- strokovnjake za delo z omrežji (predvsem za model strežnik / odjemalec);
- komunikacijami, delovnimi postajami, planiranjem kapacitete;
- strokovnjake za delo s programsko opremo, kupovanjem, nameščanjem; konfiguriranjem, uvajanjem;
- načrtovalce in upravnike sistema;
- načrtovalce in upravnike baze podatkov (DBA);

- programerje in upravnike proceduralnih predmetov;
- testno skupino, končne uporabnike in zaposlene v organizaciji, ki se dobro spoznajo na svoje delo;
- analitike za izbrana področja poslovanja, ki poznajo podatke, vedo kje so, kako jih uporabljajo in kako do njih dostopajo.

### **Določitev nalog vsem sodelujočim v projektni skupini.**

Vsak član skupine mora imeti določeno nalogo in biti zanjo odgovoren. Primeri nalog:

- zbiranje zahtev o ustreznih podatkih ter načrtovanje modela baze podatkov;
- izbira, nakup, učenje in uporaba orodij za preoblikovanje podatkov;
- izbira, nakup in učenje končne aplikacije s katero bodo delali uporabniki;
- iskanje najboljših virov podatkov;
- nameščanje, konfiguriranje in testiranje omrežja, pretvornikov in baz podatkov;
- nakup in namestitev programske in strojne opreme;
- zagotavljanje varnosti sistema;
- definiranje obnavljanja, osveževanja podatkov;
- podpora uporabnikom in njihovo uvajanje v delo.

### **Vzpostavitev arhitekture sistema in podatkov ter tehnične infrastrukture.**

Upoštevati moramo, da naj bo skladišče podatkov zasnovano kot strežnik, končni uporabniki pa kot odjemalci. Držati se moramo načela odprtih sistemov, torej se čim manj togo vezati na določenega izdelovalca, in se izogibati izdelkom, ki postavljajo preveč specifične zahteve za dodatne sestavne dele. Izbira arhitekture sistema in podatkov ter infrastrukture zelo veliko vpliva na nadaljnji razvoj skladišča podatkov.

Arhitekturo skladišča podatkov sestavlja množica pravil in navodil, ki predstavljajo okvir za načrtovanje celotnega sistema (POE, 1996). Pri arhitekturi sistema je zelo pomembna arhitektura podatkov, saj zagotavlja pretok podatkov po sistemu in s tem njihovo uporabnost v organizaciji. Vsak član projektne skupine mora razumeti diagram arhitekture skladišča podatkov. Tehnična infrastruktura je tesno povezana z arhitekturo sistema, ker obsega vse tehnologije, ki so potrebne za delovanje arhitekture sistema. Arhitektura skladišča podatkov je lahko zaradi različne infrastrukture implementirana na različne načine. Velikokrat je potrebno precej časa, da vse potrebne komponente tehnične infrastrukture spravimo skupaj, saj je večino potrebne strojne in programske opreme potrebno kupiti, namestiti, pravilno konfigurirati in se jo naučiti uporabljati.

Tehnična infrastruktura obsega:

- *načrtovanje vse potrebne in primerne strojne ter programske opreme;*

Strojna oprema mora biti odprta, podpirati mora skalabilnost, paralelnost, simetrično multiprocessing (SMP) in masivno paralelno procesiranje (MPP),

- *načrtovanje kapacitete skladišča podatkov;*

Predvideti moramo dovolj prostora na diskih, možnost priključitve drugih sredstev na katerih se bodo shranjevali podatki, zadostne količine začasnega diskovnega prostora, na katerem se bodo podatki po potrebi razvrščali in grupirali. Ker se podatki iz skladišča podatkov večinoma berejo, mnogokrat v velikih količinah naenkrat, je primerno predvideti velike medpomnilnike. V zadnjem času je potrebno predvideti tudi možnost shranjevanja slik, zvoka in drugih binarnih podatkov.

- *definiranje stopnje granularnosti oziroma podrobnosti podatkov;*
- *določitev orodij za preoblikovanje podatkov;*

To so programske rešitve, namenjene avtomatiziranju procesov izvlečevanja, čiščenja, integriranja, preoblikovanja, prenašanja in nalaganja izvornih podatkov iz heterogenih virov v

ciljne podatke v ciljnih bazah. Nekatera orodja omogočajo sinhrono ali asinhrono osveževanje analitične baze ob spremembah podatkov na operativnem nivoju,

- *definiranje načinov arhiviranja, shranjevanja varnostnih kopij (backup);*

Programska oprema mora biti čim bolj neodvisna od sredstev, na katerih se bodo shranjevali podatki in mora poleg magnetnega diska dopuščati tudi druge oblike shranjevanja, kot so optični diski, plošče CD, magnetna sredstva in drugo. Izdelava rezervnih kopij več giga ali terabajtov ni ravno enostavna, zato je to pomemben del načrtovanja skladišča podatkov,

- *definiranje načinov dostopa končnih uporabnikov do podatkov;*

Pri izbiri uporabniškega orodja nam lahko šele na koncu pomagajo končni uporabniki, saj sama izbira vmesnika na začetku nima nobenega pomena. Upoštevati moramo želje, potrebe in sposobnosti uporabnikov, saj ima izbira uporabniškega orodja veliko večji učinek na uporabo skladišča podatkov, kot bi si sprva mislili. Lahko je skladišče podatkov še tako dobro narejeno, vendar če je uporabniško orodje uporabniku neprijazno, skladišča ne bo uporabljal tako pogosto in tako dobro izkoristil,

- *definiranje načinov in frekvenc osveževanja podatkov (dnevno, tedensko, mesečno);*
- *definiranje vnaprej definiranih operacij in postopkov;*
- *definiranje komunikacijskih orodij ter mrežnih povezav LAN / WAN;*
- *definiranje baz podatkov ter povezav med njimi;*
- *definiranje pretvornikov baz podatkov (database gateway);*

Omogoča pretok podatkov med heterogenimi bazami podatkov ali sistemi. Vsebujejo povezave med različnimi omrežji, komunikacijskimi protokoli in bazami podatkov,

- *definiranje potrebne opremljenosti delovnih postaj;*
- *strokovno izobraževanje, ki je njen zelo pomemben del;*
- *definiranje sistemov za upravljanje baz podatkov (DBMS);*

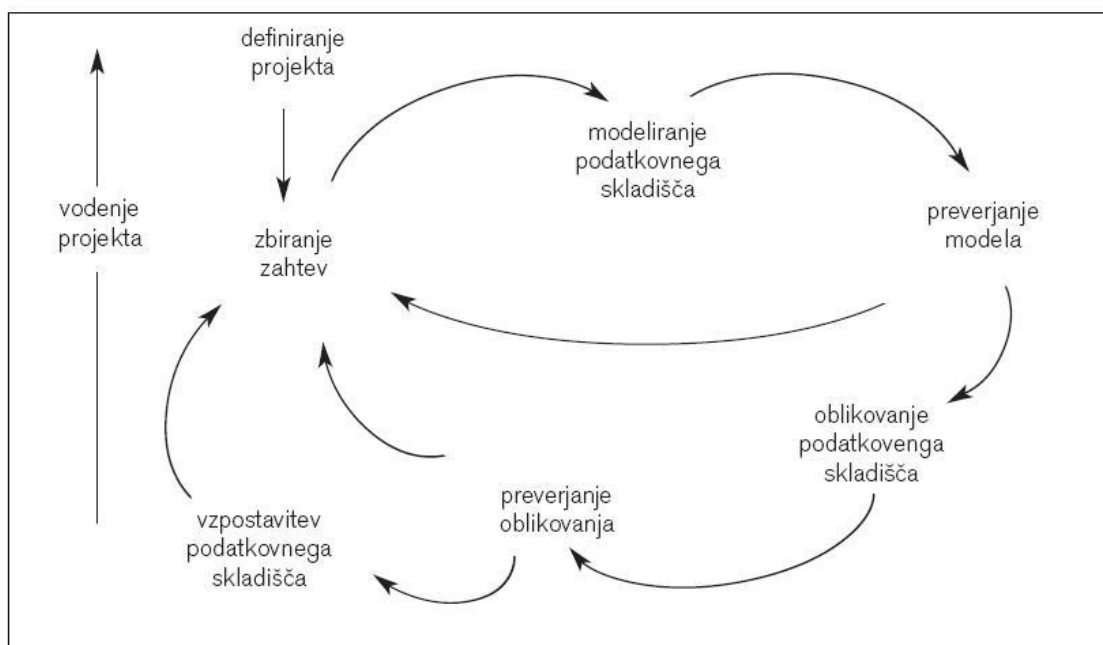
DBMS je najpomembnejši sestavni del skladišča podatkov, ki mora biti sposoben brez težav obvladati velike količine podatkov, merjeno v giga ali celo terabajtih. Namenjen je izgradnji in vzdrževanju analitične baze DBMS je lahko relacijski ali večdimenzionalni. Podpirati mora skalabilnost strojne opreme in paralelno procesiranje. Pri izboru DBMS je treba biti pozoren na odzivnost sistema pri nenehnem povečevanju podatkov, kajti količina podatkov v skladišču se bo nenehno povečevala, saj se bodo vpisovali novi podatki, stari pa se ne bodo brisali, ker je treba voditi tudi vse zgodovinske podatke (Kos, 1997, str. 46 - 48).

## 6.2 Življenjski cikel podatkovnega skladišča

Življenjski cikel se prične z definiranjem projekta, kjer se določijo cilji in namen podatkovnega skladišča. Nato sledi zbiranje zahtev. Natančno določene zahteve uporabnikov so izhodišče za modeliranje podatkovnega skladišča. Ko imamo model končan, lahko pričnemo s fazo preverjanja modela oziroma ali smo določili vsa dejstva, mere in dimenzije ter ali smo mislili na to, kakšna bo hierarhija atributov. Če ugotovimo, da je model nepopoln se faza od zbiranja zahtev in modeliranja ponovi. Ko je dimenzijski model končan, lahko pričnemo z oblikovanjem podatkovnega skladišča. To je faza fizičnega oblikovanja in definiranja tabel v podatkovni bazi. Za tem sledi preverjanje oblikovanje ter nazadnje vzpostavitev podatkovnega skladišča. Če pri preverjanju ugotovimo pomanjkljivosti se faza od zbiranja zahtev skozi celotni potek vseh faz ponovi. Pri vsem tem pa je pomembno, da je celotni projekt projektno voden.

Življenjski cikel oziroma povezanost posameznih faz življenjskega cikla podatkovnega skladišča podrobno prikazuje naslednja slika.

Slika 6: Faze razvoja življenjskega cikla



Vir: C. Ballard, D. Herrman, D. Schau, R. Bell, *Data modeling Techniques for Data Warehousing*, 1998, str. 49, slika 20.

Omenjeno je že bilo, da je razvojni cikel podatkovnega skladišča pri velikih centraliziranih podatkovnih skladiščih podatkovno voden, saj pričnemo s podatki. Vrstni red posameznih faz razvojnega cikla operativnega okolja in okolja podatkovnega skladišča pa je tako popolnoma obrnjen (Golob & Welzer, 2000).

### 6.3 Analiza zahtev za snovanje podatkovnega skladišča

Razvijalci podatkovnega skladišča morajo zbrati zahteve, jih popisati in dokumentirati. Zbrane zahteve je potrebno analizirati, to pomeni raziskati do podrobnosti. Analiza zahtev se uporablja pri izgradnji začetnega dimenzijskega modela. Vsebina analize zahtev pomeni določitev kandidatov za mere, dejstva, dimenzije, skupaj s hierarhijo dimenzij in določitev granularnosti – podrobnosti.

Zbiranje zahtev je združeno s študijo poslovnih procesov in z aktivnostmi informacijske analize. Na ta način razvijalci spoznajo in razumejo problematiko, za katero bodo izvajali modeliranje.

Zahteve za razvoj podatkovnega skladišča ne smejo biti preveč specifične, ker lahko pri oblikovanju podatkovnega skladišča izločijo dejstva, ki sicer izgledajo nepomembna, so pa ključnega pomena za analize.

Znana sta dva različna pristopa za pridobivanje poslovnih zahtev za oblikovanje podatkovnega skladišča:

- z upoštevanjem in uporabo vseh možnih podatkov iz transakcijskih sistemov: temelji na definiranju zahtev z uporabo virov podatkov v produkcijskih transakcijskih sistemih. Prednost tega pristopa je, da so znani viri podatkov, da so omejene možnosti zahtev in da se na začetku projekta porabi manj časa z uporabniki, a se poveča tveganje pojava množice nepravilnih zahtev. Ta metoda je primerna za načrtovanje celovitega podatkovnega skladišča.
- z upoštevanjem potreb in želja uporabnikov: temelji na definiranju zahtev s pomočjo raziskovanja izvajanja procesov uporabnikov. Prednosti tega pristopa so, da se med zahteve vključi tudi tisto, kar je potrebno, in ne samo tisto, kar je na voljo. Slabost pa je, da je potrebno skrbno bedeti nad pričakovanji uporabnikov. Uporabniki morajo razumeti, da se določenih podatkov ne da dobiti. Ta metoda je primernejša za načrtovanje manjših področnih podatkovnih skladišč. [W.H. Inmon, C. Ballard, M.].

Pri projektu STIRA je bila uporabljena druga metoda. Zahteve so bile zbrane s pomočjo intervjujev in krajših delovnih sestankov.

## 6.4 Dimenzijsko modeliranje

### 6.4.1 Kaj je dimenzijsko modeliranje?

Modeliranje podatkov je proces, pri katerem na podlagi zbranih zahtev naredimo ustrezen model podatkov, ki ga je nato mogoče spremeniti v fizično podatkovno strukturo. V osnovi ločimo klasično in dimenzijsko modeliranje, ki se razlikujeta v stopnji natančnosti modeliranja podatkov. Klasični podatkovni modeli so veliko bolj natančni in opisujejo entitete in povezave med njimi, ki se zrcalijo v logični strukturi baze podatkov. Poznamo več vrst modelov podatkov, najbolj pogost je entitetno-relacijski (E-R model). Poznamo še akcijske diagrame, HIPO diagrame, Jacksonove diagrame, Nassi-Schneidermanovi diagrami, navigacijski diagrami, itd.

Zaradi velikega uspeha E-R modelov so bili tudi prvi DSS in EIS sistemi narejeni z njihovo uporabo, a se je v praksi izkazalo, da omenjeni modeli kot model skladišča podatkov ni primeren, ker:

- ne omogoča dela z več zapisi istočasno,
- uporabnik se v zapletenem modelu hitro izgubi,
- je narejen za visoko normalizirane strukture
- je namenjen za veliko število transakcij, ki vsebujejo malo zapisov (OLTP). Pri skladišču podatkov je ravno obratno – malo transakcij, ki morajo obdelati veliko zapisov (OLAP),
- težko ga je spreminjati, ko je že utečen. (Kos, 1997, str 20.)

S pomočjo dimenzijskega modeliranja, ki je osnova za model skladišča podatkov, se oblikuje poslovni koncept podatkov s pomočjo uporabnikovih želja. Predstavlja nekakšno vez med informacijskim sistemom in uporabniki, saj so podatki približani miselnim procesom človeka. Kajti uporabniki so tisti, ki vedo katere informacije bodo potrebovali za odločanje. Prav nič nam pa ne koristi dobro zasnovan dimenzijski model, če nimamo nad tem primerne OLAP orodja.

Poznamo več načinov pristopa gradnje podatkovnega skladišča, in sicer;

- od zgoraj navzdol: model pričnemo graditi tako, da se najprej zberejo zahteve uporabnikov ter šele nato poiščemo ustrezne podatkovne vire, ki jih bomo prenesli v podatkovno skladišče.
- od spodaj navzgor, kjer je proces ravno obraten, saj začnemo pri virih podatkov.

Uporabimo lahko oba načina, tako da vse podatke z čim manjšim naporom stresemo v skladišče. Pri tem nastane smetišče podatkov, vendar imajo uporabniki pred sabo neko osnovno različico skladišča podatkov in si lažje predstavljajo za kaj pravzaprav gre ter tako lažje definirajo, kaj potrebujejo. Na osnovi tako dobljenih uporabniških zahtev je z iterativno metodo lažje zgraditi pravi model podatkov. Iz svojih izkušenj se mi zdi kombiniran pristop koristnejši kot vsak zase. V praksi se mi je zdelo najbolje uporabiti princip od zgoraj navzdol ter tako na podlagi uporabniških zahtev poiskati vire podatkov. Včasih je nato koristno poiskati v določenih virih podatke (notranji ali zunanji), ki bi jim mogoče koristili. Ko uporabniki vidijo določeni prototip analiz se jim včasih porodijo nove ideje, ki zahtevajo tudi dograjevanje ali spreminjanje dimenzijskega modela. To se ponavlja toliko časa, dokler uporabniki ne potrdijo določene rešitve, ki jim jo nato predamo v uporabo.

Uporabniki transakcijskega sistema se razlikujejo od uporabnikov, ki uporabljajo podatkovna skladišča. OLTP uporabniki so zaščiteni pred bazo podatkov preko uporabniškega vmesnika. Izvajajo fiksne operacije, ki so del transakcijskega delovnega procesa. Aplikacije podatkovnega skladišča so popolnoma drugačne, saj niso procesno orientirane, temveč podatkovno. Uporabniki ponavadi razpolagajo direktno s podatki, kjer ponavadi ni definiranih fiksnih delovnih procesov (z občasnimi izjemami). Končni uporabniki se ne zavzemajo za kopičenje podatkov v podatkovnem skladišču (s transakcijami), temveč želijo iz njega potegniti informacije. Nad podatki iščejo odgovore na svoja vprašanja, testirajo in preverjajo hipoteze, določajo si možne dogodke, odkrivajo vse mogoče vzorce in primerjave glede na določeno časovno obdobje, raziskujejo in projektirajo za prihodnost. (Ballard, C., et al, str. 82).

Dimenzijski model je pravi način modeliranja podatkov za podatkovna skladišča, saj v veliki večini primerov dosežemo cilje podatkovnega skladišča.

Zakaj sploh potrebujemo večdimenzionalen pristop? Kakšne koristi pričakujemo? Zakaj nismo zadovoljni kar z rezultati poizvedb iz Excela, Accesa in podobnih orodij? Glavni problem teh orodij je, da nam ne omogočajo hierarhičnih, večdimenzionalnih agregatov in izvajanje večdimenzijskih kalkulacij (Thomsen, 1997). Današnji končni uporabnik potrebuje za izvajanje zahtevnih analiz, ki vključujejo izračune in primerjavo podatkov vzdolž dimenzij, hiter in zanesljiv sistem. Sistemi, ki takšne analize podpirajo, temeljijo na osnovi večdimenzionalnih podatkovnih baz. Postopek načrtovanja večdimenzionalnih lahko strnemo v naslednje korake: Spoznavanje trenutne situacije, definiranje kock, dimenzij in povezav, definiranje hierarhij in članov dimenzij ter definiranje formul.

## 6.4.2 Tipi in uporaba dimenzijskih modelov

Podatki morajo biti organizirani na takšen način, kot jih želijo videti odločevalci, ker se poskuša v podpori odločanju predstaviti informacije na preprost način ter omogočiti uporabniku enostavno in hitro spreminjanje tabel, tako da le-te dobijo v dani situaciji večji pomen. Preprosto pa morajo biti strukturirani tudi zato, ker se dela z ogromnimi količinami podatkov. V podatkovnih skladiščih, namenjenih podpori odločanju, se zato uporabljajo



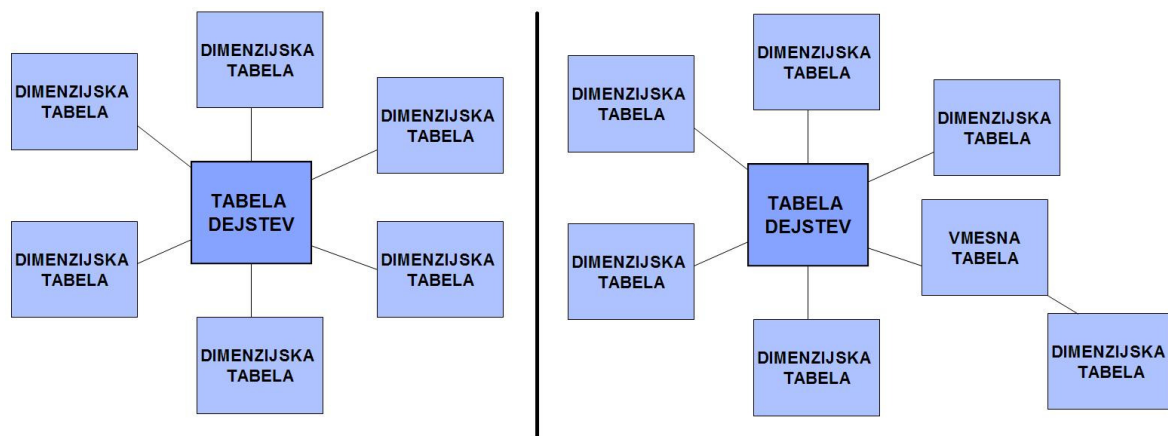
večdimenzionalni podatkovni modeli. Bistvena značilnost teh modelov je, da so sheme, katere lahko realiziramo, preproste in blizu načinu razmišljanja odločevalca (Jaklič, 2002, str 105; OLAP, 2004).

Dimenzijski model je množica povezanih tabel v podatkovnem skladišču. Tabela dejstev je osrednja tabela, nanjo pa je vezanih več tabel dimenzij.

Poznamo dve obliki sheme:

- **shema v obliki zvezde** omogoča hiter dostop do podatkov, ker je s tabelo dejstev neposredno povezana vsaka tabela dimenzije;
- **shema v obliki snežinke** je lažja za vzdrževanje, ker so tabele normalizirane. Shema prikazuje spodnja slika.

Slika 7: Shema Zvezde in Snežinke



Vir: Avtor

Vsaka tabela vsebuje primarni ključ. Glavni ključ tabele dejstev je sestavljen iz enega ali več tujih ključev, ki so primarni ključi v dimenzijskih tabelah.

Osnovna ideja dimenzijskega modeliranja izhaja iz tega, da je skoraj vsak tip poslovnih podatkov mogoče predstaviti v obliki dimenzijske kocke, kjer polja pomenijo mere, robovi kocke pa dimenzije. Kimball (2007) večdimenzijsko kocko, ki vsebuje več kot tri dimenzije, imenuje tudi »hiper kocka« (angl. *hypercube*). Ponavadi modeli vsebujejo od 4 do 15 dimenzij. Več kot 20 dimenzij ni priporočljivo, saj se z večanjem dimenzij povečuje tudi nepreglednost modela.

Dimenzijski model vključuje pri zvezdasti strukturi le dve vrsti tabel, in sicer glavne in dimenzijske tabele, medtem ki ima shema snežinke še vmesne tabele, ki deloma normalizirajo podatkovni model. Nekatera olap orodja podpirajo le zvezdasto strukturo. Zvezdasta struktura ima z razliko od strukture snežinke preprostejši model ter s tem enostavnejše razumevanje celotnega modela kot tudi meta podatkov. Ima dobro definirane relacije med tabelami ter omogoča tudi enostavnejše definiranje podrejenosti in nadrejenosti posameznih atributov v dimenzijah.

Če želimo razumeti dimenzijski model, moramo najprej poznati, kaj pomenijo omenjeni pojmi:

**Tabela dejstev:** tabela dejstev je glavna osrednja tabela, ki je popolnoma normalizirana, vsebuje kvantitativne podatke o poslovanju oz. **mere**. Lahko bi jo poimenovali tudi tabela mer (angl. *Fact table*), saj vsebuje mere oziroma dejstva in spremenljivke, po katerih lahko te mere ocenimo. Navadno so to številčni podatki, bodisi decimalna števila, cela števila ali celo konstante, če jih uporabniki potrebujejo. Karakteristike glavnih tabel so, da rešujejo povezave mnogo proti mnogo med dimenzijskimi tabelami ter so ponavadi 90 % večje kot dimenzijske tabele.

**Dimenzijska tabela:** Če tabela dejstev vključuje mere, ki jih želimo spremljati, nam dimenzijska tabela pove, kaj vse želimo analizirati v okviru tabele dejstev. Dimenzijska tabela vsebuje spremenljivke in pridevke v opisni obliki. Dimenzija je lahko sestavljena iz ene ali več hierarhij, ki omogočajo podrobnejši vpogled so denormalizirane ter povezane z glavnimi tabelami.

**Dimenzije** so tekstovni atributi, ki opisujejo stvari in so organizirani znotraj dimenzij. Pri modeliranju je potrebna pozornost glede vrste dimenzij. Poznane so:

- počasi spreminjajoče dimenzije: glavna predpostavka je, da se ključ dimenzije ne spremeni;
- hitro se spreminjajoče male dimenzije: pomembno je učinkovito sledenje na pridevku, saj je včasih težko določiti, kaj je hitra in kaj počasna sprememba;
- velike dimenzije (npr. 100 milijonov stavkov): obvladovanje takih dimenzij je možno s pravilnim indeksiranjem. Najslabši primer je hitro se spreminjajoča velika dimenzija. Problem se reši tako, da se kreira dve dimenziji glede na hitrost spreminjanja atributov. Primer take dimenzije je Prebivalec, ki samo v ZDA vsebuje 100 milijonov zapisov;
- degenerirane dimenzije: degenerirane dimenzije imajo ključ, nimajo pa atributov. To se zgodi v primeru, ko zapisujemo individualne podatke v tabelo mer, ki se na primer nanašajo na enega prebivalca in so »skrite« v tabeli mer. (Paulin, 2005 & Ravbar, 2003).

**Hierarhija elementov:** je logično urejanje dimenzijskih elementov do najvišje ali najnižje stopnje podrobnosti. Lahko bi si jo predstavljali kot primer odnosa starš – otrok, kjer otrok pomeni nižjo stopnjo podrobnosti ali granulacijo prednika. Tako stopnja hierarhije pomeni možnost dodatnega raziskovanja po podatkih. Višja stopnja vsebuje tudi večjo stopnjo sumarizacije in obratno. Pri raziskovanju podatkov, kjer določamo stopnjo agregacije sta se uveljavila angleška izraza. Tako raziskovanju ali **vrtnanju** navzdol pravimo drill-down ter navzgor roll-up.

**Agregiranje podatkov:** je proces združevanja določenih podatkov. Omogoča nam zniževanje stopnje granularnosti v dimenzijskem modelu.

Poznamo več vrst struktur, ki se med seboj ločijo po načinu definiranja ključev in po številu glavnih tabel. Tako poznamo: popolnoma normalizirano shemo, delno denormalizirano shemo, popolnoma denormalizirano shemo (Tabela 2). Primerjave in lastnosti shem so razvidne iz spodnje preglednice.

Tabela 2: Primerjava in tipi shem

Tipi shem	Strukture lookup tabel	Prednosti	Slabosti
Popolnoma normalizirana shema	Vsebuje atributov ID in opis stolpca, vključno z ID stolpca njegovega starša (nadrejene tabele – parent table)	Minimalna zasedenost prostora v podatkovnem skladišču in minimalna redundanca podatkov	Zahteva mnogo povezav (joinov) za pridobitev informacij od najvišjega nivoja do lookup tabel.
Delno denormalizirana shema	Vsebuje atributov ID in opis stolpca, vključno z ID stolpca njegovega starša ter predhodnega starša (angl. <i>Grandparent table</i> )	Bistveno zmanjša število povezav, ki so potrebne za povezavo atributa z starim staršem (angl. <i>Grandparent table</i> )	Zahteva nekaj dodatnega prostora
Popolnoma denormalizirana shema	Vsebuje atributov ID in opis stolpca, vključno z ID ter opisom stolpca njegovega starša ter predhodnega starša (angl. <i>Grandparent table</i> )	Nadalje zmanjša potreben povezave za pridobitev atributovega opisa	Zahteva veliko prostora v podatkovnem skladišču

Vir: *Microstrategy, Microstrategy Project Design, str. 72*

Dimenzijski model mora v celoti odražati poslovni proces, kot ga razumejo managerji in torej odsevajo poslovne potrebe ter omogočajo odgovor na vsa poslovna vprašanja. Če govorimo z jezikom managerjev in uporabnikov, govorimo o poslovni uspešnosti, stroških, govorimo o merah. Elemente, katere z merami merimo, pa imenujemo dimenzije (Hughes, 2002, str 295).

**Metapodatki:** Metapodatki so podatki o podatkih. Vsebujejo informacije o strukturi podatkov in o povezavah med strukturami podatkov znotraj baze podatkov ali pa med več bazami podatkov. Meta podatki so poseben del skladišča podatkov, ki razvijalca seznanijo z vsebino skladišča podatkov, da razume: kaj pomeni podatek, od kod izvira, v kakšnem formatu je zapisan, število korakov potrebnih za preoblikovanje, stopnjo in način agregacije, časovna razporeditev prenosov, arhiviranja, itd. V grobem ločimo operativne metapodatke, ki opisujejo povezave med viri in objekti v analitični bazi in transformacije podatkov. Ti meta podatki so opredeljeni kot tehnični metapodatki. Drugi so znani kot DSS metapodatki, ki podatke skladišča podatkov preslikajo v uporabniku lažje razumljiv poslovni dimenzijski model. Opisujejo povezave med objekti v analitični bazi in v večdimenzijskem modelu.

### 6.4.3 Načrtovanje večdimenzijskega modela

Pred začetkom načrtovanja večdimenzionalne podatkovne baze moramo najprej spoznati in razumeti trenutno situacijo obravnavanega sistema, za katerega želimo ponuditi ustrezno rešitev. Obstoječe stanje obravnavamo z logičnega in fizičnega vidika, pri čemer v fizičnem vidiku proučimo predvsem obstoječe sistemske rešitve, v logičnem vidiku pa poslovne procese in pravila. Rezultat analize obstoječega stanja je logičen model sistema na visokem nivoju, kjer opredelimo vse izvore podatkov in uporabnike (Lahajnar, 2000, str 102).

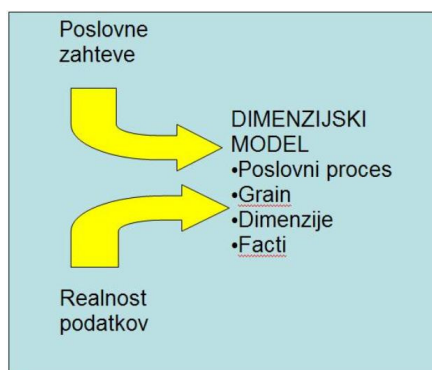
Pri načrtovanju večdimenzijskega modela je potrebno:

- določiti tabele dejstev;
- določiti dimenzijske tabele: v okviru teh table moramo določiti dimenzijske oziroma deskriptivne attribute ter primarne ključe, ki v tabeli dejstev predstavljajo tuji ključ. V zvezdasti strukturi so dimenzijski atributi v isti tabeli kot dimenzijski elementi, medtem ko so v strukturi snežinke shranjeni v ločenih, ki se povezujejo z dimenzijsko tabelo;
- ko definiramo dimenzijsko tabelo, si moramo zamisliti, kakšna bo hierarhija elementov oziroma po čem bomo raziskovali podatke navzgor ali navzdol;
- določiti stopnjo granularnosti oziroma podrobnosti podatkov. To je ena najpomembnejših faz pri načrtovanju dimenzijskega modela. Višja je stopnja podrobnosti tem podrobnejši so kasnejši prikazi le teh. Nizka stopnja granularnosti omogoče predvsem prikaze agregiranih podatkov;
- Določiti mere, ki predstavljajo predvsem preračunane vrednosti za dane transakcije;
- določiti ključe, povezave in odnose med tabelami dejstev in dimenzijskimi tabelami.
- določiti, katere operacije se bodo morale izvesti, da bomo dobili nove vrednosti podatkov;
- določiti možne agregate;
- testiranje, preverjanje in izpopolnjevanje dimenzijskega modela.

Novembra 2007 sem se udeležila petdnevnega tečaja Dimensional Modeling in Depth, ki sta ga je vodila priznana strokovnjaka s področja podatkovnih skladišč, in sicer Ralph Kimball ter Margy Ross. Predavatelja imata zelo široko teoretično ter praktično znanje in sta zelo strokovno odgovarjala na zastavljena vprašanja. V nadaljevanju bom predstavila kratka pomembna izhodišča.

Pred načrtovanjem dimenzijskega modela se moramo vprašati ali imamo glede na zahteve uporabnikov oziroma glede na njihove želje tudi razpoložljive in obstoječe podatke (Slika 8).

Slika 8: Vhodi za dimenzijsko modeliranje



Vir: R. Kimball & M. Ross, *Dimensional modeling in Depth*, 2007.

Cilji vsakega skladišča podatkov so:

- zmanjšati čas polnjenja podatkov v podatkovno skladišče: več glede polnjenja podatkovnega skladišča bo opisano v nadaljevanju pod poglavjem Polnjenje podatkovnega skladišča;
- zmanjšati čas razvoja podatkovnega skladišča ter
- sprejemljiv dostop do podatkov: včasih ne moremo glede na kompleksnost sprotnih poizvedb zagotoviti takojšnjega dostopa do podatkov. Vendar kljub temu časi ne smejo biti predolgi, saj ima to lahko za posledico odpor uporabnikov do uporabe podatkovnega skladišča.

Ko se odločamo za tip dimenzijskega modela, se pravi ali shema snežinke ali zvezde, se odločimo če se le da za zvezdasto strukturo, saj nam v snežinki dodatne povezave zmanjšujejo performance sistema. Shema zvezde denormalizira dimenzijski model med tem ko ga shema snežinke normalizira. Kadar imamo ogromno količino povezav mnogo proti mnogo v dimenzijskih tabelah, je seveda shema snežinke upravičena. Kadar količina podatkov ni prevelika, je tudi vseeno, če tabele vsebujejo redundantne podatke. Dodatni podatki nam sicer zasedejo več prostora v bazi podatkov, vendar nič zato, če s tem pridobimo na odzivnih časih. Podatkovno skladišča so namenjena za shranjevanje velikih količin podatkov.

Omenjene so bile tudi lastnosti posameznih dimenzijskih tabel, in sicer:

- Datumske dimenzije: ni se nam treba bati novih in nepričakovanih datumov. Ko definiramo datumske dimenzije je odvisno od projekta za katerega se uporablja. Kimball predlaga, da če se le da, naj za celotno datumsko oziroma koledarsko dimenzijo obstaja čim manj tabel, saj bi s tem pridobili tudi na odzivnih časih.
- Generične dimenzije: pomeni, da je sama dimenzijska tabela razdeljena na podnivoje. Omenjenih dimenzij naj se raje izogibamo, ker zmanjšujejo preglednost modela.
- Junk dimenzije; v eni tabeli imamo več podatkov (npr. številka naročila, način plačila, pogoje plačila, način prevoza,...). Uporabili naj bi jo samo v primeru, kadar iščemo določeno korelacijo med podatki, drugače se jim raje izognimo.
- Ujemajoče se dimenzije (angl. *Conformed Dimensions*): že ime pove, da so to dimenzije, ki se ujemajo. Vsebujejo določene enake zapise. Tak primer sta tabeli leta in meseci, kjer tabela meseca lahko vsebuje tudi zapis leta.
- Počasi se spreminjajoče dimenzije. Poznamo 3 tipe omenjenih dimenzij. Več bo opisano pod naslednjim poglavjem.

Ko se načrtuje dimenzijski model, se v določeni situaciji vprašamo ali uporabiti eno ali dve dimenzijski tabeli. Kadar imamo povezavo m:1 ponavadi uporabimo eno dimenzijsko tabelo, razen v določenih izjemah kot so naprimer: pogoste spremembe ali velika količina podatkov (> milijon). Kadar imamo povezavo m:m je priporočljiva uporaba več dimenzijskih tabel v kombinacijo z vmesno tabelo. V primeru, da je količina podatkov majhna pa jih zaradi odzivnih časov lahko združujemo.

Pri načrtovanju dimenzijskega modela je po Kimballu (2007) priporočljiva uporaba matrike oz. kot jo sam imenuje Data Warehouse Bus Matrix. Primer prikazuje spodnja tabela. Pred tem določimo:

- poslovni proces;
- poslovno podrobnost (angl. *Grain*): predstavljati si moramo, kaj nam ene vrstica v tabeli dejstev predstavlja. Nova granularnost pomeni novo fact tabelo;
- dimenzije: opisne vsebine, ki se povezujejo z fact tabelo;
- numerične mere (ang. Facts), ki odražajo vrstico v tabeli dejstev (ang Fact table) oziroma poslovno podrobnost.

*Tabela 3: Primer matrike po Kimball-u*

Poslovni proces	DIMENZIJE			
	Datum	Produkt	Stranka	Dobavitelj
Prodaja	X	X	X	
Nabava....	X	X		X

*Vir: R. Kimball & M. Ross, Dimensional modeling in Depth, 2007.*

V zgornjem primeru bi tako imeli 2 fact tabeli, in sicer ena za področje prodaje, druga za področje nabave, ki bi imeli določene skupne dimenzije. To so: datum, produkt in dobavitelj.

#### 6.4.4 Počasi spreminjajoče dimenzije

V knjigi The Data Warehouse Toolkit Kimball govori o klasičnem načinu obvladovanja počasi se spreminjajoče dimenzije (Kimball, 1998, str 180) kot je lahko geografska dimenzija. Glavna značilnost omenjenih dimenzij je, da se ne spreminja šifra, ki označuje posamezni pridevek, temveč opis pridevka, ki se navezuje na to šifro. Problem se pojavi pri izdelavi poizvedb, ki se nanašajo za več časovnih obdobj, saj jih med seboj težko primerjamo. Kimball predlaga tri tipe možnosti omenjeni dimenzij.

1. Slowly Changing Dimension: poznamo 3 tipe počasi spreminjajočih dimenzij, in sicer:
  - a. TIP1: Spremenimo vrednost atributa iz stare na novo vrednost. Pri tem se moramo zavedati, da s tem izgubimo zgodovino vrednosti za nazaj.
  - b. TIP2: V dimenzijsko tabelo dodamo novo vrstico z novim nadomestnim ključem. S tem ohranimo zgodovino, vendar povečamo kompleksnost poročanja. Priporočljivo je, da dodatnemu polju dodamo tudi končni datum veljavnosti zapisa. Je eden najbolj priporočljivih načinov.
  - c.

- d. TIP3: Dodamo nov stolpec v dimenzijsko tabelo. Stari opis pridevka prenesemo v novo polje, novi zapis pa v staro polje, kjer se je nahajal prejšnji opis pridevka. Tako se nam poročila, ki vsebujejo poizvedbe po teh podatkih spremenijo in osvežijo z novimi opisi. Najmanj pogosto uporabljena metoda.

Prvi tip se ponavadi uporablja v primeru napak, ki nastanejo bodisi zaradi prenosa ali kakega drugega razloga. Tako stare vrednosti nima smisla hraniti v dimenzijski tabeli. Omenjeni način sem sama v praksi uporabila za primer opisa nekega statusa, ker so uporabniki zahtevali drugačno poimenovanje. Drugi tip je primeren predvsem za sledenje sprememb opisov pridevkov v dimenziji. Pri drugem tipu se moramo zavedati, da pridevki vsebujejo do določenega časovnega obdobja neko vrednost in nato drugačno. V praksi sem sama omenjeni problem reševala z definiranjem filtrov oziroma pogojev na samih poročilih glede na datum nastanka. V določenih primerih pa tudi uporabniki zahtevajo, da se prikazujejo stare in nove vrednosti zapisov. Kimball priporoča uporabo pri velikih in sestavljenih dimenzijah, kjer je veliko sprememb. Tak primer je lahko dimenzija Stranka in Proizvod, ki vključuje veliko zapisov. Stranko kot osebo lahko vodimo od njenega nastanka, preko sprememb v njenih atributih, do njene ukinitve ali izbrisa iz registra (Kimball, 1998, str 180). Na primeru Zavoda bi omenjeni način lahko prišel prav tudi pri vodenju zavarovancev ali izvajalcev zdravstvenih storitev. Pri tretjem načinu nas sledenje ne zanima, prav tako ne moremo enostavno spremeniti zgodovine kot je to možno pri drugem tipu. Določeno mero lahko spremljamo na stari ali novi način. Omenjena metoda je uporabna v statistiki.

Obstajajo tudi kombinacije med prvim in drugim tipom ter med drugim in tretjim tipom. Kombinacije so odvisne od posameznih odločitev, kaj se želijo uporabniki spremljati, katere zapise potrebujemo in katere ne, kaj želimo slediti in podobno.

#### **6.4.5 Kako v zasnovi modela upoštevati performance sistema?**

Ko se postavlja podatkovno skladišče, moramo že pred tem strmeti, da bodo odzivni časi čim krajši oziroma, da bodo uporabniki najhitreje prišli, do zelenih prikazov podatkov. Agregiranje podatkov je eden najpomembnejših razlogov za izboljšanje odzivnega časa poizvedb ter s tem zmanjšanje obremenjenosti sistema, zlasti enote CPU.

K performancam sistema lahko pripomore tudi način, kako je dimenzijski model zgrajen. Kimball zagovarja pristop, če se le da, naj se izbere zvezdnata shema, saj bolj zapletena shema snežinke vpliva na slabše uporabniško razumevanje in vzdrževanje ter podaljša odzivne čase.

Zvezdasta struktura pa lahko postane neprimerna, če so dimenzijske tabele zelo velike (primerljivo s številom zapisov v glavni tabeli), kar neugodno deluje na performanse ali v primeru, ki je shema preveč denormalizirana, kar zahteva preveč diskovnega prostora in s tem prevelike stroške. Z drugimi besedami – hitrost preiskovanja narekuje čim bolj enostavno strukturo, optimiranje kapacitet diskov pa čim bolj normalno podatkovno strukturo. (Kastelic 1997).

Performance oziroma odzivnost sistema lahko dosežemo tudi s pomočjo dobrega poznavanja administracije baz podatkov. Že samo s pravilnim pristopom kreiranja posameznih indeksov lahko bistveno izboljšamo delovanje aplikacij.

Odzivnosti skladišča podatkov ne moremo izboljševati v nedogled, ampak se ustavimo pri nekem vnaprej določenem odzivnem času, ki je še sprejemljiv za uporabnike in dosegljiv za razvijalce in vzdrževalce. Izboljševanje odzivnosti je lahko neskončen proces, ki traja dokler živi sistem, s tem pa so povezani tudi stroški. Slaba odzivnost ne povzroča le slabe volje uporabnikov, ampak vpliva tudi na sprejemanje poslovnih odločitev in s tem na učinkovitost poslovanja podjetja, saj uporabniki morda ne bodo mogli pravočasno ali dovolj natančno opraviti svojega dela. Na splošno velja, da je odzivni čas skladišča podatkov sprejemljiv takrat, ko omogoča uporabnikom, da svoje delo opravijo v predvidenem času (Ferle, 1996).

Nekateri dejavniki, ki vplivajo na odzivnost so:

- veliko število manjših računalnikov, med katerimi so naloge porazdeljene;
- računalniško omrežje, s katerim so povezani ti računalniki;
- stalno povečevanje količine podatkov in števila uporabnikov skladišča podatkov.

Odzivnost skladišča podatkov se lahko izboljša z:

- razporejanjem obremenitev skladišča: obremenitve je smiselno razporediti na čas, ko skladišče uporablja manj uporabnikov;
- dokupovanjem strojne in / ali programske opreme: pred nakupom zmogljivejše strojne opreme, ki je povsem sprejemljiva rešitev vendar povezana s precejšnjimi stroški, je smiselno odzivnost izboljšati programsko. Tak način pa prej ali slej doseže zgornjo mejo zmogljivosti računalniškega sistema in takrat je potrebno kupiti zmogljivejšo strojno opremo ali pa zamenjati del programske opreme.

## 7 Polnjenje podatkovnega skladišča

### 7.1 ETL proces

ETL (angl. *Extract, Transform, Load*) je eden najpomembnejših in zahtevnejših procesov, ki skupaj s podatkovnim skladiščem tvori srce vsakega sistema za poslovno obveščanje. Po drugi strani pa je najbolj zapostavljen in v večini primerov »glavni« krivec za neuspešen projekt implementacije. Zapostavljen je predvsem v smislu nepoznavanja zmožnosti v očeh naročnika in implementatorja (NPS, 2008).

Že sama kratica pove, da gre za zajem, transformacijo in prenos podatkov v podatkovno skladišče. V samo implementacijo ETL postopkov je potrebno vložiti veliko naporov in v praksi velja, da pri celotni izgradnji skladišča podatkov, zavzamejo ETL postopki lahko tudi do 70 % porabljenega časa.

Proces ETL zajema naslednje tri korake, in sicer:

- **Ekstrakcija podatkov** – je proces pridobivanja podatkov iz notranjih poslovnih sistemov ali zunanjih virov. V tem koraku pridobimo podatke, ki jih bomo uporabili v procesu transformacije. Določimo orodja in tehnike za zajem podatkov.
- **Transformacija podatkov** – je proces transformacije in čiščenja ter priprave podatkov za podatkovno skladišče. Transformacija lahko vključuje spremembe podatkovnih tipov, imena stolpcev, izločevanje slabih podatkov, dopolnjevanje podatkov (npr s časovnim atributom, naslovom, imenom...), popravki določenih podatkov, razreševanje nepravilnih podatkov, itd. Za določene večdimenzionalne analize se lahko tudi združuje ter s tem do določene mere zmanjša količino podatkov. Ko podatke spustimo skozi proces transformacije, sledi kontrola preverjanja natančnosti virov podatkov, da se potrdi ali podatki odražajo realno stanje.



- **Prenos podatkov** – je zadnji korak, ko podatke prenesemo v podatkovno skladišče in so pripravljeni za analitične poizvedbe. Določimo orodja in tehnike za nalaganje podatkov v podatkovno skladišče.

Kimball zagovarja 2 načina polnjenja, in sicer podatke lahko istočasno, ko jih pridobivamo spustimo skozi proces transformacije ali jih najprej prečrpamo v ločeno okolje in nato nad njimi izvajamo proces transformacije (Kimball, 2007).

Podatke lahko pridobimo:

- iz notranjih poslovnih sistemov (sistemi, ki obstajajo v podjetju) ali
- iz zunanjih virov (razni javno dostopni registri, podatki na oddaljenih lokacijah, itd.).

## 7.2 Problemi pri polnjenju podatkovnih skladišč

Veliko naporov je potrebno vložiti v določitev najboljše oziroma najustreznejše metodologije in nato v samo implementacijo ETL postopkov. Nesporno je implementacija ETL postopkov eden izmed glavnih rizičnih dejavnikov uspeha projekta. Ovire, kot so nečisti ali pomanjkljivi podatki, neskladna poslovna pravila, omejena razpoložljivost izvornih podatkov, časovni intervali izvajanja, itd. Velikokrat pripeljejo projekt čez časovne in denarne okvire. Zato morajo biti postopki polnjenja zasnovani pregledno ter rezultate o uspešnem oziroma neuspešnem poteku ustrezno dokumentirati. In prav sodobna ETL orodja to omogočajo. Dobra dokumentacija je v veliko pomoč administratorju podatkovnega skladišča. Proces polnjenja je avtomatiziran in se izvaja v določenih časovnih intervalih.

## 7.3 Načini polnjenja podatkovnih skladišč

Procese ETL lahko izvajamo:

- ročno (npr. s pomočjo poizvedovalnih jezikov: naprimer: SQL procedur ali s pomočjo pomočjo programskih jezikov, kjer za specifični primer postopka ETL napišemo ustrezn program;
- s pomočjo ETL orodij, ki jih je na trgu kar nekaj (Informatica, DataStage, DWE,...). ETL orodje je priporočljivo pri prenosu in transformacije ogromne količine podatkov. Nudijo paleto gradnikov, kjer lahko definiramo proces polnjenja.

Sodobna ETL orodja nam omogočajo pomoč pri naslednjih fazah:

- priključitev na poljuben vir ne glede na izvor podatkov (poljubne podatkovne zbirke podatkov, poljubne poslovne aplikacije, nestrukturirani izvori – razne datoteke);
- preverjanje kvalitete podatkov in združevanje podatkov iz več sistemov (npr. poenotiti šifrant kupcev preko več sistemov z vgrajenimi matematičnimi algoritmi za odkrivanje ujemanja podatkov);
- hitrost prenosa podatkov (zmožnost paralelnega delovanja omogoča prenose ogromnih količin podatkov v kratkem obdobju);
- pregled izvora in celotnega tokokroga podatkov, ki bo postal v prihodnosti nujen (NPS, 2008);
- zapletene in enostavne transformacije podatkov, agregacije, itd.;
- omogočanje avtomatičnega izvajanja procesov polnjenja;
- dober pregled nastalih napak v procesu polnjenja, itd.;

Predvsem so omenjena vrsta orodij investicija za prihodnost, ker se ni potrebno spraševati ali bo mogoče dobiti podatke ali lahko ugotovimo izvor podatkov ali bomo dovolj hitro napolnili podatkovno skladišče ter kako bomo implementirali kvaliteto podatkov. Prav tako je potrebno pogledati po svetu in videti, da gre sam trend hitrosti in želje analize podatkov iz dnevnega osveževanja v t.i. realno časovno osveževanje in tukaj glavno vlogo odigrajo ravno opisana orodja. (NPS, 2008)

## 8 Orodja poslovne inteligence

Ko so se začeli razvijati sistemi za podporo odločanja, managerski in direktorski informacijski sistemi, je bil poudarek na zagotavljanju zadostne količine podatkov in informacij za odločanje. V tem času pa je bilo na voljo le malo podatkov, kateri bi bili primerni za računalniško obdelavo, strojna oprema in programi za njihovo obdelavo pa niso bili dovolj zmogljivi. Ko pa so podatki v digitalni obliki naraščali, se je problem premajhne količine podatkov spremenil v problem zagotavljanja pravih podatkov – podatke katere managerji zares potrebujejo. Odločevalci so imeli tako vse več podatkov, tudi nepotrebnih, potrebovali pa so tudi vse več poročil, katere zaradi raznolikosti odločitvenih situacij ni bilo mogoče pripraviti v naprej. Za poizvedovanje po podatkovnih bazah se je uporabljal jezik SQL (angl. *Structured query Language*), ki je postajal vse bolj enostaven za uporabo. Managerji bi lahko s pomočjo SQL-a prišli do pravih informacij, vendar ima ta jezik zaradi svoje narave preprostega jezika le omejene možnosti. Zato z njimi ne moremo, poiskati odgovorov na vsa poslovna vprašanja. Zaradi vse večjih potreb po podatkih, pa tudi ker informacijske službe niso uspele dovolj hitro zagotoviti novih poročil in pogledov na podatke, so se razvila orodja OLAP, katera omogočajo managerjem, da si sami enostavno izdelajo poljubne poglede na podatke in dostopajo neposredno do virov podatkov. Da pa lahko managerji ta orodja uspešno uporabljajo, morajo biti enostavna za uporabo ter mora biti podatkovni vir primerno pripravljen (Jaklič, 2002).

Orodja OLAP so orodja poslovne inteligence, ki so bila v magistrski nalogi že omenjena. Omogočajo večdimenzijske analize podatkov, ki nadomestijo več zaporednih poizvedb in uporabniku olajšajo navigacijo in vrtanje po podatkih.

OLAP orodja se delijo na večdimenzionalna (MOLAP – Multidimensional On-line Analytical Processing), katera delujejo v večdimenzionalnih podatkovnih skladiščih, relacijska (ROLAP – Relational On-Line Analytical Processing), ki omogočajo neposreden dostop do podatkov v relacijskih bazah podatkov in pa na hibridna (HOLAP – Hybrid On-Line Analytical Processing), ki so kombinacija prejšnjih dveh (Berson et al, 1999, str. 69).

Tipične operacije, ki se izvajajo z orodji OLAP (Jaklič, 1999):

- zvižanje (angl. *roll-up*) – omogoča manj podroben opis podatkov;
- vrtanje v globino (angl. *drill-down*) – omogoča bolj podroben opis podatkov;
- rezanje (angl. *slice in dice*) – omogoča iskanje v določeni kategoriji podatkov;
- vrtenje (angl. *pivot*) – omogoča različne poglede na podatke.

Podatki v podatkovnem skladišču so prečiščeni ter agregirani ter so najboljši vir za zapletene analize preteklega poslovanja ter načrtovanje prihodnjega.

## 9 Orodje Microstrategy za delo nad podatkovnimi skladišči

MicroStrategy je obsežna, zmogljiva in dostopna platforma poslovne inteligence, ki omogoča vpogled v poslovne informacije s pomočjo zanesljivih in utemeljenih analiz poslovnih informacij. Pomaga upravljati ali kontrolirati številna možna področja poslovanja ter s tem prispeva k boljši učinkovitosti poslovanja. Predpogoj za to je dobro zasnovan koncept podatkovnega skladišča ter dobra kvaliteta podatkov.

Orodje MicroStrategy je analitično orodje za delo nad podatkovnimi skladišči, ki so zasnovana na osnovi večdimenzijskega modela. Omogoča povezavo do različnih podatkovnih baz preko ODBC povezave.

### 9.1 Paleta izdelkov, ki sestavlja Microstrategy platformo

Microstrategy platforma poslovne inteligence je močna, zmogljiva in dostopna platforma, ki omogoča vpogled v poslovne informacije s pomočjo zanesljivih in utemeljenih analiz. Predpogoj za to je dobro zasnovan koncept podatkovnega skladišča ter kvaliteta podatkov.

Orodje Microstrategy je torej analitično orodje za delo nad podatkovnimi skladišči. Podatkovno skladišče je zasnovano na osnovi večdimenzijskega modela.

Uporabniki za dostop do poročil lahko uporabljajo Microstrategy desktop ali Microstrategy Web. Razlika je v tem, da v drugem primeru uporabnik za svoje interaktivne analize uporablja poljuben spletni brskalnik. Oba omogočata pregledovanje poročil, vrtanje v globino, pripravi ad-hoc poročil in kreiranja novih poročil (nekateri funkcionalnosti so odvisne od pravic uporabnika).

Microstrategy platformo je sestavljena iz naslednjih produktov, ki so na kratko predstavljeni v nadaljevanju:

Orodje vključuje 2 serverja, in sicer:

**Microstrategy Intelligence Server** – močno zmogljiv strežnik poslovne inteligence, ki uporablja storitveno usmerjeno arhitekturo.

**Microstrategy Narrowcast Server** – je pravzaprav informacijsko dostavljalski strežnik, ki omogoča prenos poslovnih informacij uporabnikom preko mailov, tiskalnikov, datotečnih storitev, sms-jev in mobilnih naprav.

Moduli orodja Microstrategy prikazuje naslednja preglednica. V grobem jih delimo na:

- storitvene module;
- uporabniške vmesnike ter
- razvojna orodja.

Tabela omenjenih modulov je prikazana v nadaljevanju.

Tabela 4: Moduli orodja Microstrategy

Storitveni moduli	Uporabniški vmesniki	Razvojna orodja
<b><u>MicroStrategy OLAP Services</u></b> – je razširitev Microstrategy Intelligent Serverja, ki omogoča uporabnikom Desktopa, Office-a in Web-a, da manipulirajo z inteligentnimi kockami.	<b><u>Microstrategy Desktop</u></b> – je BI komponenta, ki omogoča integrirane query-je, poročanje, močna analitika in podpora procesom odločanja	<b><u>Microstrategy Administrator</u></b> – skupek orodij za administracijo orodja.
<b><u>Microstrategy Report services</u></b> – funkcionalnost Microstrategy platforme, ki omogoča, da je celoten sklop poročil dostavljen uporabnikom	<b><u>Microstrategy Mobile</u></b> - Interaktiven vmesnik, ki omogoča poslovnim uporabnikom dostopati do poročil s pomočjo BlackBerry naprav.	<b><u>Microstrategy Architect</u></b> – hitro razvojno orodje, ki mapira fizično strukturo podatkovne baze v logični poslovni model ter jih kot meta podatke shranjuje v Metabazo (Metadata Repository).
<b><u>Microstrategy Data Mining Services</u></b> – popolnoma integrirana komponenta, ki omogoča uporabnikom rudarjenje s pomočjo modelov predvidevanj.	<b><u>Microstrategy Office</u></b> – uporabnikom je omogočen dostop do uporabe funkcionalnosti celotnih Office-ovih aplikacij.	<b><u>Microstrategy BI Developer Kit</u></b> – začetno orodje z aplikacijskimi moduli in razvojnim orodjem, ki omogoča uporabnikom seznanitev s celotnim razvojem poročanja in analiz.
<b><u>Microstrategy SAP Services</u></b> – SAP NetWeaver, integracija z SAP BW, SAP WAS & SAP Enterprise Portal	<b><u>Microstrategy Web</u></b> – je visoko interaktiven uporabniški vmesnik (uporabljamo lahko Internet Explorer), preprost za uporabo, namenjen poročanju, monitoring-u in analizam	<b><u>Microstrategy Integrity Manager</u></b> – orodje za podatkovno validacijo, avtomatsko primerja in preveri podatke na poročilih ter tako poišče spremembe, ki so nastale v BI sistemu
		<b><u>Microstrategy SDK</u></b> – je obsežno razvojno okolje, ki omogoča hitro in enostavno kustomizacijo in integracija Microstrategy platforme s katero drugo platformo

Vir: P. Schulze Klaus P, Microstrategy, Zapiski tečaja (2005)

## 9.2 Opis funkcionalnosti orodja

MicroStrategy arhitektura je povezana s petimi tipi poslovne inteligence, ki so dostopni preko enotnega spletnega vmesnika, ki izboljšuje poslovno poročanje in analiziranje podatkov:

1. **Poslovno poročanje (Enterprise Reporting):** omogoča izpolnjevanje raznolikih zahtev od celovitih pregledov poslovanja pripravljenih za vodilne do obsežnejših zahtev naprednih uporabnikov.

2. **Analiza kock (MOLAP):** Zmožnost izvajanja OLAP analiz na predefiniranih podmnožicah podatkov. Uporabniki opazijo povečano hitrost in povezljivost pri delu z definiranimi podmnožicami podatkov.
3. **Ad-Hoc poizvedbe (ROLAP):** Uporaba OLAP funkcionalnosti pri izvajanju poizvedovalnih analiz in ad-hoc poizvedb, ki omogočajo vrtenje v katerikoli smeri, page-by, vrtenje podatkov, skupne vsote, sortiranje, filtriranje, rangiranje in izvažanje. Analitikom in managementu to omogoča podrobnejši vpogled v poslovne rezultate.
4. **Statistične analize in odkrivanje zakonitosti v podatkih ( angl. Data Mining)** Zmožnost uporabe statističnih postopkov in 'data mining' algoritmov za pripravo napovedi. Samo MicroStrategy orodja za poslovno inteligenco omogočajo uporabnikom izdelati analizo, ki sicer zahteva dodatne programske pakete.
5. **Pošiljanje poročil in opozarjanje (Alerting):** Možnost razpošiljanja poročil in opozoril za celotno podjetje, z možnostjo samostojnega naročanja na poročila ali z uporabo skupne administracije. Poročila se dostavi na poljuben elektronski medij, vključujoč e-pošto, tiskalnik, tekstovno sporočilo SMS, dlančnik, fax, pozivnik, ...

Zadnji dve področji trenutno v Zavodu nista podprti, sta pa uporabni in omogočeni z dokupom dodatnih licenc.

Kratek opis uporabe orodja je priložen v prilogi.

### 9.3 Dinamične nadzorne plošče

Dinamične nadzorne plošče (angl. *Dynamic Enterprise Dashboards*) so funkcionalnost tehnologije MicroStrategy Report Services. Novost je bila izdana v verziji MicroStrategy 8.1.

Funkcionalnost je podprta v verziji Microstrategy 8.1 in nam omogoča enostavno izgradnjo uporabnih nadzornih plošč ter poročil sistema kazalnikov (Scorecards) v standardni AJAX in novi Flash tehnološki razširitvi v MicroStrategy.

»Dynamic Dashboard« so dinamične nadzorne plošče, ki združujejo interaktivnost in dober vizualni vpogled v podatke.

#### Nadzorne plošče omogočajo:

- enostaven dostop in uporabo zbranih poslovnih podatkov ter lažji vpogled v sledenje oziroma spremljanje poslovanja;
- dinamično integriran vpogled v skupne podatke organizacije s pomočjo napredne vizualne predstavitve podatkov;
- prispevek k optimizaciji procesov odločanja;
- poslovni uporabniki lahko vidijo (ali sami oblikujejo) sebi prilagojene vsebine in dober vizualni vpogled v želene analize kot so: grafi, tabele, zemljevidi, prikaz indikatorjev trendov in še mnogo drugega;
- več različnih vizualnih prikazov na eni strani;
- orodje vključuje standardno AJAX in novo Flash tehnologijo ter s tem uporaba napredne vizualne funkcionalnosti;
- omogočena je razširitev na Adobe Flex Builder 2, ki še razširi možnosti prikazov podatkov na veliko možnih načinov;

- grafi v nadzornih ploščah (najboljša uporaba tipov grafov, določanje lastnosti, optimizacija predlog nadzornih plošč, hiter preklop v drugo obliko grafa ali preglednice, itd.);
- Oblikovanje plasti oziroma jezičkov za preglednejši prikaz, oblikovanje analitične nadzorne plošče, prikaz različnih sklopov podatkov, oblikovanje naslov,...

Spodnja slika prikazuje nekaj praktičnih primerov iz prakse.

Slika 9: Primer dinamičnih nadzornih plošč



Vir: Jamnik, M. (2007). Dynamic Enterprise Dashboards, str. 1-9.

V Zavodu se omenjena funkcionalnost do sedaj še ni uporabljala. Dinamične nadzorne plošče so nova napredna funkcionalnost, ki jo bo Zavod glede na dejavnost s katero se ukvarja, lahko s pridom izkoristil.

## 10 Kratka predstavitev Zavoda za zdravstveno zavarovanje Slovenije

### 10.1 Poslanstvo in vizija Zavoda

Zavod posluje preko svojih 10 območnih enot s 45 izpostavami. Vsaka območna enota s svojim delovanjem pokriva določeno območje. V njenem okviru pa obstaja mreža izpostav, ki poslujejo v posameznih občinah. Za področje informacijske dejavnosti je ustanovljena samostojna področna enota. Takšna poslovna mreža zagotavlja, da je zavarovalna storitev v največji meri približana zavarovancem. Obstajajo naslednje območne enote:

- OE Celje (Laško, Šmarje pri Jelšah, Slovenske Konjice, Žalec, Šentjur pri Celju),
- OE Koper (Izola, Piran, Ilirska Bistrica, Postojna, Sežana),
- OE Krško (Brežice, Sevnica),
- OE Kranj (Jesenice, Škofja Loka, Radovljica, Tržič),
- OE Ljubljana (Cerknica, Domžale, Grosuplje, Hrastnik, Idrija, Kamnik, Kočevje, Litija, Logatec, Ribnica, Trbovlje, Vrhnika, Zagorje),
- OE Maribor (Ormož, Lenart, Ptuj, Slovenska Bistrica),
- OE Murska Sobota (Gornja Radgona, Lendava, Ljutomer),
- OE Nova Gorica (Ajdovščina, Tolmin),
- OE Novo mesto (Črnomelj, Metlika, Trebnje),
- OE Ravne na Koroškem (Mozirje, Slovenj Gradec, Radlje ob Dravi, Velenje)

Zavod za zdravstveno zavarovanje Slovenije (v nadaljevanju: Zavod) je bil ustanovljen 1. marca 1992. Zavod izvaja obvezno zdravstveno zavarovanje s prispevki, ki jih plačujejo z zakonom opredeljeni zavezanci. Prvi razvojni dokument, ki je bil namenjen predvsem konstituiranju sistema zdravstvenega zavarovanja oz. Zavoda, je bil izdelan že v letu 1994, v začetku leta 1997 pa je Zavod na osnovi obsežnih analiz pripravil nov razvojni dokument - Strateški razvojni program zdravstvenega zavarovanja v R Sloveniji (v nadaljevanju Razvojni program). Le-ta je bil metodološko zasnovan kot projektno usmerjen proces, v katerem so bile predstavljene strategije za graditev učinkovitega in uspešnega sistema zdravstvenega zavarovanja ter v strateškem projektne planu predvideni projekti, potrebni za njihovo uresničitev. V letu 2002 je bil izdelan in sprejet nov strateški razvojni program z naslovom "Tradicija solidarnosti in spremembe po meri zavarovancev". Osrednja značilnost tega razvojnega programa je bila postavitve zavarovanca v središče razvojnih prizadevanj. Zaradi novih okoliščin (Konvergenčni program Vlade RS, nakazane reforme na več področjih) je skupščina Zavoda v letu 2006 sprejela Dopolnitve Strateškega razvojnega programa Zavoda za leti 2006 in 2007, ki je v elektronski obliki dostopno tudi preko spletnih strani. 24.09.2008 pa je Skupščina ZZZS sprejela nov Strateški razvojni program ZZZS za obdobje 2008 do 2013.

Zavod je od sprejema prvega Razvojnega programa pa do danes izvedel številne razvojne aktivnosti in projekte, med katerimi velja z vidika razvoja Zavoda in zdravstvenega zavarovanja nekatere še posebej izpostaviti:

Z ustanovitvijo Vzajemne zdravstvene zavarovalnice d.v.z. konec l. 1999 je Zavod izvajanje prostovoljnih zdravstvenih zavarovanj, ki so se do tedaj izvajala na Zavodu, prenesel na novo družbo. Z ustanovitvijo Vzajemne je bilo veliko pozornosti namenjene tudi delitvi prej skupnih informacijskih sistemov – delitvi evidenc osebnih podatkov in pooblastil za dostop do posameznih sklopov baz podatkov.

V letu 2000 je Zavod dokončno uvedel sodobno elektronsko kartico zdravstvenega zavarovanja (KZZ) v slovensko zdravstvo, ki je za zavarovance pomenila hitrejši in prijaznejši vstop v zdravstveni sistem in hkrati novo fazo v razvoju sodobnega in enovitega zdravstveno informacijskega sistema v Sloveniji. Z namenom poenostavitve in preglednejših postopkov obveznega zdravstvenega zavarovanja so bili v obdobju 2002-2006 na tem področju izvedeni še nekateri projekti, ki so omogočili nove uporabnosti kartice – zapis izdanih zdravil, medicinsko tehničnih pripomočkov ter podatkov o darovalcu organov na kartico. Zaradi številnih poslovnih in tehničnih razlogov se je Zavod v letu 2006 odločil za prenovo kartičnega sistema, in sicer postopno v smeri uvedbe neposrednih dostopov do podatkov, ki se sedaj nahajajo na kartici. Enostavno, kakovostno in učinkovito prenašanje podatkov ter komuniciranje v okviru on-line zdravstvenega zavarovanja je namreč ena temeljnih razvojnih usmeritev Zavoda v prihodnjih dveh letih, ki jo bo Zavod skušal uresničiti z izvedbo dveh projektov, ki bosta usmerjena v uvedbo on-line sistema ter razvoj nove kartice zdravstvenega zavarovanja, profesionalne kartice in infrastrukture javnih ključev. S tem se bodo za zavarovane osebe poenostavili postopki uresničevanja pravic iz zdravstvenega zavarovanja, izvajalcem zdravstvenih storitev bodo na voljo bolj točni in ažurni osebni in medicinski podatki, kar bo povečalo kakovost zdravstvenih storitev, zdravstvenim zavarovalnicam pa omogočilo racionalizacijo stroškov poslovanja. Povezava na več informacij... Aktivnosti na področju kartice zdravstvenega zavarovanja pa v zadnjih letih zaznamuje tudi intenzivno sodelovanje Zavoda v mednarodnih dejavnostih in projektih, kot posledica vključenosti Slovenije v Evropsko unijo.

Kot pobudnik projekta za uvedbo zavarovanja za dolgotrajno nego je Zavod maja 2004 izdelal podroben predlog za uvedbo tega zavarovanja, ki bi sistemsko in dolgoročno uredil problematiko dolgotrajne nege v Sloveniji. Dolgotrajna nega je namreč eno od stičnih področij socialnega in zdravstvenega varstva, ki v Sloveniji še ni evropsko primerljivo urejeno. Predlog je bil posredovan ministrstvu za zdravje in ministrstvu za delo, družino in socialne zadeve z namenom, da se pripravi in sprejme ustrezna zakonodaja. Na področju zdravil in medicinsko tehničnih pripomočkov je Zavod s pogovori in pogajanjem uspel znižati cene vrsti zdravil in medicinsko tehničnih pripomočkov. Pozitivni rezultati so še posebej vidni na področju zdravil, kjer je Zavod poleg vsakoletnih pogajanj o znižanju cen, v letu 2003 uvedel še način zagotavljanja pravic do zdravil na podlagi njihove zamenljivosti. V letu 2005 je bila vzpostavljena nova baza zdravil, ki je postala nepogrešljiva evidenca za delo lekarn, Zavoda in Inštituta za varovanje zdravja. Aktivnosti in prizadevanja na področju zdravil je Zavod podkrepil še z izvedbo nacionalnega projekta za promocijo varne in pravilne rabe zdravil, ki je bil usmerjen predvsem v pripravo in izvedbo obsežne medijske kampanje ter množično izdajo tiskanih gradiv za promocijo varne in pravilne rabe zdravil - S sodelovanjem Zavoda v nacionalnem projektu »Razvoj upravljanja sistema zdravstvenega varstva«, ki je potekal pod okriljem Ministrstva za zdravje in sodelujočih ustanov ob sofinanciranju Svetovne banke, je bil junija 2004 uveden nov obračunski model na področju akutne bolnišnične obravnave, ki predstavlja racionalnejšo podlago za načrtovanje bolnišnične zdravstvene dejavnosti. V ta namen so bile izvedene tudi spremembe v sistemu računalniškega izmenjevanja obračunskih dokumentov med izvajalci zdravstvenih storitev in Zavodom ter spremembe v programski opremi za evidentiranje in kontrolo teh dokumentov Na Zavodu.

Z namenom temeljite prenove informacijske podpore beleženju in kontroli opravljenih in Zavodu zaračunanih zdravstvenih storitev za potrebe spremljanja stroškov po izvajalcih, dejavnostih in zavarovanih osebah, bo Zavod predvidoma do spomladi l. 2009 izvajal poseben projekt, v okviru katerega bodo podatki s specifikacij organizirani na način zbirk podatkov ter vzpostavljene pilotne informacijske rešitve za podporo analizam podatkov o opravljenih zdravstvenih storitvah v akutni bolnišnični obravnavi (SPP) in za zdravila, izdana na recept (AOR).

Intenzivna prizadevanja Zavoda so bila v zadnjih treh letih usmerjena tudi v organizacijsko prenovo in uvedbo informacijske podpore področju uveljavljanja regresnih zahtevkov. Razvita in uvedena je bila celovita informacijska podpora izvajanju poslovnega procesa za pridobivanje, zbiranje in evidentiranje podatkov o nastali škodi in uveljavljanje povračil škode, do katerih ima Zavod pravico povračila. Vgrajene kontrole zagotavljajo ažurno, točno in enotno vodenje podatkov o nastalih škodah in s tem dobro osnovo za učinkovito uveljavljanje regresnih zahtevkov v okviru območnih enot in Zavoda kot celote. Organizacijska prenova in uvedba informacijske podpore področju pomeni za Zavod dodatnih cca. 20 mio evrov prihodkov na letni ravni, kar je med drugim tudi posledica spremembe 86. člena Zakona o zdravstvenem varstvu in zdravstvenem zavarovanju (november 2003) ter uveljavitve novele Zakona o obveznih zavarovanjih v prometu (april 2006).

Zavod od jeseni 2006 dalje načrtuje in izvaja v sodelovanju s partnerji v zdravstvu ter izvajalci zdravstvenih storitev obsežen projekt uvedbe on-line sistema in prenove kartice zdravstvenega zavarovanja, katerega končni cilj je popoln on-line sistem, v katerem prenovljena kartica zdravstvenega zavarovanja in nova profesionalna kartica ne nosita podatkov, pač pa samo digitalna potrdila, na podlagi katerih je mogoč neposreden, varen in



zanesljiv dostop do podatkov, ki se nahajajo na strežnikih ZZZS in prostovoljnih zdravstvenih zavarovalnic. (ZZZS, 2009)

## 10.2 Vizija razvoja informacijskega centra

V Strateškem načrtu razvoja informacijskega sistema v Zavodu je opredeljena vizija te področne enote, ki se glasi:

Področna enota Informacijski center bo zagotavljala informacijsko infrastrukturo na področju zdravstvenega zavarovanja v Sloveniji in bo delovala kot povezovalac razvoja informatizacije v sistemu zdravstvenega varstva na nacionalnem nivoju ter v skladu z možnostmi tudi za nekatera področja socialne varnosti.

Strateški razvojni načrt med drugim vsebuje poudarke: stalno sodelovanje na projektih, ki se izvajajo po planu Strateškega razvojnega načrta Zavoda, varovanje osebnih podatkov, izvajanje varnostne politike, stalno povečevanje razpoložljivosti informacijskega sistema, postavitvev "Disaster Recovery" plana in postopno vzpostavitev rezervne lokacije. Informatika Zavoda aktivno sodeluje pri izgrajevanju rešitev skladno s strategijo Ministrstva za zdravje, e-Zdravje.

Strategija razvoja storitev za prihajajoče obdobje upošteva dosežen razvoj informatizacije v Sloveniji in infrastrukturne pogoje, ki so danes v Sloveniji na razpolago:

- telekomunikacijska infrastruktura, ki omogoča zanesljiv prenos večje količine informacij kadarkoli in kjerkoli v državi,
- infrastruktura javnih ključev, ki omogoča varen prenos informacij,
- vzpostavljene nacionalne zbirke podatkov.

Kot posledica dolgoletnega sodelovanja z različnimi partnerji s področja zdravstva in informatike je v Področni enoti Informacijski center zbran največji del domačega znanja s področja integriranih informacijskih rešitev za podporo zdravstvenega zavarovanja ter celotnega sistema zdravstvenega varstva.

Nekaj prvenstvenih razvojnih področij predstavljamo v naslednjih odstavkih.

### Upravljanje odnosov s strankami - CRM

Zavod za zdravstveno zavarovanje Slovenije z razvojnimi projekti na področju informatike spodbuja in omogoča razvoj sodobnih oblik medsebojnega komuniciranja z vsemi partnerji in s tem prispeva h kakovosti poslovnih odnosov in zniževanju stroškov. Sodobne rešitve temeljijo na elektronskem izmenjevanju podatkov, storitvah z uporabo kartice zdravstvenega zavarovanja ter e-storitvah.

Zavodovo poslovno okolje sestavljajo:

- Okoli 2 milijona zavarovanih oseb, vključenih v obvezno zdravstveno zavarovanje,
- Preko 845.000 zavezancev za plačilo prispevkov obveznega zdravstvenega zavarovanja,
- preko 1.600 pogodbenih izvajalcev zdravstvenih storitev ter
- drugi javni zavodi in institucije kot so Ministrstvo za zdravje, Ministrstvo za javno upravo, Zavod za pokojninsko in invalidsko zavarovanje, Zavod RS za zaposlovanje, Statistični urad RS, in drugi.

## Rešitve za zavarovane osebe

Zavarovane osebe so ključni uporabniki storitev Zavoda, zato je njim namenjen sodoben, informacijsko podprt servis, ki je predpogoj za kakovostne storitve.

Kartica zdravstvenega zavarovanja omogoča enostavnejšo uporabo zdravstvenih storitev in boljšo preglednost nad veljavnostjo obveznega zdravstvenega zavarovanja. Spodbuja kakovost zbirk podatkov pri zavezancih, izvajalcih zdravstvenih storitev in na Zavodu. Zavarovanim osebam je na voljo uporaba 285 samopostrežnih terminalov po vsej Sloveniji, kjer lahko potrjujejo kartice, pridobivajo informacije o pravicah iz obveznega zdravstvenega zavarovanja, o ponudbah zdravstvenih storitev ter storitev zdravstvenega zavarovanja. Zavarovanci lahko zastavljajo vprašanja, na katera dobijo strokovne odgovore.

Na samopostrežnih terminalih in spletnih straneh Zavoda je možno naročanje konvencijskih potrdil za morebitno uporabo nujnih medicinskih storitev v tujini, evropskih kartic zdravstvenega zavarovanja in evropskih potrdil E111.

## Izmenjevanje podatkov z izvajalci zdravstvenih storitev

Že od leta 1996 leto se uporabljajo in rastejo rešitve računalniškega izmenjevanja podatkov med izvajalci zdravstvenih storitev in Zavodom. Najbolj uveljavljeno področje je izmenjevanje računov in zahtevkov za plačilo, ki jih za opravljene zdravstvene storitve izvajalci pošiljajo na Zavod, z izvajalci zdravstvenih storitev na primarni ravni pri posredovanju podatkov o izbranih osebnih zdravnikih in z lekarnami pri posredovanju podatkov o izdanih zdravilih na recept, z dobavitelji medicinsko tehničnih pripomočkov o izdanih medicinsko tehničnih pripomočkih. Podrobna navodila za vključitev v računalniško izmenjavo podatkov so na voljo tudi v elektronski obliki.

Izvajalci lahko prek Zavodovih spletnih strani brezplačno pridobivajo e-Zavodove publikacije in e-navodila namesto delno plačljivih papirnih izdaj le teh.

## Elektronske storitve Zavoda

Zavod je med prvimi začel z uporabo novih orodij za razvoj varnih spletnih storitev za naše poslovne partnerje. Ena prvih razvitih spletnih storitev je možnost nadzorovanega vpogleda / posredovanja osebnih podatkov pooblaščenim osebam organizacij z ustreznimi pravnimi podlagami (e-Poizvedbe). S to spletno rešitvijo je Zavod razbremenil referente prijavno odjavnih služb, saj na Zavod letno pride več kot 200.000 zahtev po posredovanju podatkov raznim državnim organom, institucijam, sodiščem, itd.

Zavod je kot nosilec prijav / odjav / sprememb v obvezno zdravstveno zavarovanje, obvezno invalidsko zavarovanje, zavarovanje za primer brezposelnosti osrednji nosilec in izvajalec za vse navedene institucije in se je kot partner vključil v sistem priprave celovitih elektronskih rešitev znotraj sistema e-VEM.

## Drugi javni zavodi in institucije

V skladu z zakonskimi podlagami in medsebojnimi pogodbami Zavod elektronsko posluje z nekaterimi javnimi zavodi in institucijami. Zavod v elektronski obliki zagotavlja podatke iz prijav v obvezno zdravstveno, pokojninsko in invalidsko zavarovanje Zavodu za pokojninsko in invalidsko zavarovanje, Zavodu za zaposlovanje, Statističnemu uradu in drugim zakonsko

določenim upravičencem do podatkov na osnovi poizvedb. S strani Ministrstva za notranje zadeve in Statističnega urada pridobiva Zavod podatke iz nacionalnih registrov. Iz Davčne uprave pridobiva podatke o izterjavi prispevkov obveznega zdravstvenega zavarovanja. Zelo dejavno je elektronsko posredovanje podatkov tudi z Ministrstvom za zdravje in Inštitutom za varovanje zdravja na različnih področjih.

Vse komponente informacijskega sistema zavoda se stalno nadgrajujejo in s stalnim spremljanjem novih tehnoloških in organizacijskih možnosti informacijski center zagotavlja tudi v bodoče varno in sodobno elektronsko poslovanje

### **Podpora finančnemu poslovanju**

Zavod za zdravstveno zavarovanje Slovenije upravlja z eno največjih blagajn v Sloveniji. Zato posveča vedno več pozornosti obvladovanju prihodkov in odhodkov za zdravstveno zavarovanje. V te namene je bil vzpostavljen centraliziran informacijski sistem SAP za knjiženje, finančno poslovanje in kontroling. Doma je bila razvita računalniška aplikacija v podporo izterjavam škodnih zahtevkov pri poškodbah po tretji osebi in pri poškodbah pri delu. V razvoju sta še dva projekta za obvladovanje finančnih sredstev: prenova in posodobitev poslovnega področja s pomočjo IKT za prispevke ter za izdatke zdravstvenih storitev. Prvi projekt bo v pomoč plačevanju, nadzoru in izterjevanju prispevkov. Drugi projekt bo poleg obvladovanja izdatkov za zdravstvene storitve omogočal kvalitetnejše analiziranje porabe finančnih sredstev za obvezno zdravstveno zavarovanje ter lažje planiranje razporejanja le-teh.

### **Intranetne rešitve**

Uspešno interno komuniciranje, kakovostno upravljanje z znanjem zaposlenih in enotno izvajanje poslovnih procesov so ključni izzivi za uspešne intranetne rešitve. Za razvoj to vrstnih rešitev so bila izbrana orodja Lotus Notes /Domino.

Razvita je bila široka paleta aplikacij, ki omogočajo enostavnejše in bolj pregledno izvajanje poslovnih procesov:

- dokumentne zbirke,
- rešitve za podporo skupinskemu delu in
- rešitve za interno komuniciranje.

Z aplikacijami dokumentnih zbirk se urejeno vodijo manjše zbirke podatkov na različnih področjih: stiki z javnostmi, informacijsko-dokumentacijska dejavnost, baze znanja, projektna in druga razvojna dokumentacija, projektne pisarne, zapisniki organov upravljanja Zavoda, urejanje spletnih strani, manjše evidence idr.

Podpori skupinskemu delu in internemu komuniciranju so namenjene rešitve službe za pomoč uporabnikom informacijskega sistema, evidentiranja in spremljanja prejetih računov, rezervacije skupnih virov, evidentiranja in potrjevanja odsotnosti z dela, diskusijske baze in številne druge rešitve.

### **Upravljanje kakovosti in uvajanje najboljših praks informacijsko komunikacijske tehnologije**

Področna enota Informacijski center se načrtno ukvarja z razvojem upravljanja kakovosti informacijskega sistema. Področja, kjer je kakovost še posebej pomembna so: razvoj programske opreme, njeno uvajanje in vzdrževanje, nabava strojne in programske opreme, vsakodnevni operativni postopki, varovanje in zaščita, upravljanje s človeškimi viri, splošna administracija.

V ta namen Področna enota informacijski center neposredno sodeluje s poslovnimi področji in vodstvom Zavoda, sodeluje pri zasledovanju poslovnih ciljev in omogoča hitro analiziranje poslovanja. Uporablja preizkušene metodologije in pristope.

Za pomoč uporabnikom je znotraj Področne enote Informacijski center organizirana služba HELP DESK-a, ki sprejema klice o incidentih in izvede ali organizira njihovo reševanje.

V informacijskem centru je vložen trud na vseh področjih, ki dvigujejo kakovost informacijskega sistema in s tem prispevajo k dvigu kakovosti poslovanja v Zavodu. Poseben trud je vložen za permanentno vzdrževanje in dopolnjevanje internih standardov, dopolnjevanje postopkov in njihovo izboljševanje. Posebna skrb je namenjena izobraževanju zaposlenih informatikov z dodatnimi znanji s poslovnih področij, na katerih delajo. K povečanju kakovosti prispevajo tudi redni postopki interne revizijske službe. Običajni so tudi periodični pregledi zunanjih revizorjev. Na osnovi priporočil revizorjev se izvajajo nekatere razvojne naloge ali razvojni projekti, ki ob realizaciji prispevajo k višji ravni kakovosti v Zavodu.

### **10.3 Organizacijska struktura Zavoda**

Organigram mezoorganizacijske strukture direkcije in poslovne enote informacijski center je priložen v prilogi.

### **10.4 Kratek opis direktorskega informacijskega sistema**

ZZZS DIS je aplikacija, ki omogoča spremljanje ključnih poslovnih parametrov s področja nadzora zavarovancev. Vsebuje podatke iz obveznega zavarovanja, pripravljene na način, ki omogoča hitre in enostavne analize teh ključnih parametrov (ZZZS, 2002, str 5).

V preteklosti ni bilo velikega uspeha pri gradnji celovitega podatkovnega skladišča oziroma direktorskega informacijskega sistema. Omenjeno podatkovno skladišče še vedno obstaja in se polni mesečno iz transakcijskih sistemov. Glavni problem za slabši interes s strani uporabnikov je preobseženo načrtovan sistem ter veliko boljših analiz omogočenih s strani produkcijskih javanskih aplikacij.

### **10.5 Cilji zagotavljanja podpore na področju podatkovnega skladiščenja v Zavodu**

Cilj je zagotoviti analitično poročanje na zahtevo za aktualno poslovno problematiko na različnih poslovnih področjih. V prihodnje naj bi se gradile manjše obvladljive podatkovne

tržnice, ki bodo enostavnejše z vidika načrtovanja ter kasneje vzdrževanja delujoče rešitve. Ko podatkovno skladišče predamo v uporabo, ne smemo zaspati. Kasneje se lahko spremenijo določeni podatki, za kar je potrebno izvesti določene popravke.

Cilj načrtovanja podatkovnih tržnic je, da bi odločanje potekalo še boljše in učinkovitejše kot do sedaj.

## **11 Izgradnja podatkovnega skladišča v Zavodu za primer projekta STIRA (Predpisovanje zdravniških receptov po izvajalcih in zdravnikih glede na dejavnost)**

### **11.1 Kratak opis področja STIRA (predpisovanje receptov po zdravnikih)**

Dosedanje spremljanje podatkov o predpisanih receptih za zdravnike in izvajalce poteka po ustaljenih poteh, ki so se izoblikovale tekom let (začetek je bil v letu 2000), ko je bila metodologija STIRA sprejeta in se je uveljavila kot sistem seznanjanja zdravnikov in izvajalcev s predpisovanjem receptov in z letno ciljno vrednostjo za recepte, ki naj bi jo zdravniki dosegli, da bi bilo njihovo delovanje oziroma predpisovanje receptov racionalno in ekonomsko upravičeno.

Obdelava podatkov STIRA in informiranja Zavodovih izvajalcev je proces, katerega lastnik je Področje za razvoj zdravstvenega zavarovanja (PZRZZ).

Na kratko obdelava podatkov STIRA z oznako zajema prikaz podatkov o predpisanih receptih po izvajalcih in zdravnikih, glede na število predpisanih receptov pozitivne (P) in vmesne liste (V), število opredeljenih oseb in pogojnih enot opredeljenih oseb (za zdravnike z izbirami), vrednosti receptov in letne ciljne vrednosti za zdravnika ter indeksa odstopanja dejanske od ciljne vrednosti za določeno obdobje. Omenjena obdelava podatkov se izvaja s pomočjo PL programov.

Letno se izvedejo, štiri četrletne kumulativne obdelave podatkov o predpisanih receptih, kjer se izvajalce kot ključne nosilce informiranja njihovih pogodbenih partnerjev seznanijo z rezultati obdelave za celotno leto. Sprejeto spremljanje predpisovanja zdravil po izvajalcih z modelom za določitev planiranih oziroma ciljnih vrednosti za zdravila predpisana na recept za posameznega izvajalca je hkrati podlaga za presojo racionalnosti predpisovanja zdravil in ni vzvod sankcioniranja ne izvajalcev niti ne zdravnikov. Skladno s tem je vsakemu izvajalcu – pogodbenemu partnerju določen ciljni znesek, do katerega lahko predpisuje zdravila na recepte na račun zdravstvenega zavarovanja. V nadaljevanju je na kratko predstavljena metodologija oziroma kako pridemo do izračuna ciljnih vrednosti, ki nastanejo zaradi načina spremljanja podatkov o receptih in drugih podatkov, ki so zajeti v računalniško obdelavo, saj so le te nujen pogoj za interpretiranje rezultatov.

## **11.2. Izhodišča in kriteriji za izračun ciljnih vrednosti po metodologiji STIRA**

Za uresničevanje projekta je najprej potrebno zagotoviti dostopnost in ažurnost iz baze podatkov o izbiri osebnega zdravnika, iz baze izvajalcev ter izdanih zdravilih predpisanih na recept oziroma baze receptov.

Podlaga za merila in ciljne vrednosti so vrednosti receptov, ki so jih zdravniki sami predpisali v preteklem letu. Te vrednosti so ustrezno korigirane s pričakovano realno rastjo odhodkov za zdravila v obveznem zavarovanju in pričakovano inflacijo v prihodnjem letu. Za posamezne skupine zdravnikov (po specialnostih) so izračunane letne ciljne vrednosti. Seštevek ciljnih vrednosti vseh zdravnikov zaposlenih pri posameznem izvajalcu predstavlja ciljno vrednost tega izvajalca. Vir podatkov za spremljanje predpisovanja zdravil po izvajalcih so podatki iz avtomatske obdelave receptov, podatki o številu in strukturi opredeljenih oseb pri zdravnikih iz baze izbira osebnega zdravnika ter podatki o zdravnikih po posameznih specialnostih iz baze podatkov izvajalcev.

Na podlagi podatkov o predpisanih receptih v preteklem letu, podatkov UMAR o pričakovani inflaciji v Sloveniji v tekočem letu ter planirano rastjo odhodkov obveznega zavarovanja za zdravila in lekarniško dejavnost v tekočem letu, se določijo letne ciljne vrednosti za zdravnike za posamezne specialnosti (ZZZS, 2008).

Kriteriji za določitev ciljnih vrednosti so opredeljeni posebej za:

- splošne zdravnike v osnovni zdravstveni dejavnosti,
- ginekologe na primarnem nivoju,
- specialiste skupine 1 (ambulantno delo),
- specialiste skupine 2 (režija, diagnostika),
- zobozdravnike,
- maksilofacialno kirurgijo.

### **11.2.1 Metodologija ciljnih vrednosti**

Ciljne vrednosti so določene za posamezne skupine zdravstvenih dejavnosti. Za osnovno zdravstveno dejavnost in ginekologijo na primarnem nivoju so ciljne vrednosti izračunane v povezavi z izbiro zdravnika oz. osebnega ginekologa, kar omogoča standardizacijo oz. primerljivost med zdravniki z različnimi starostnimi in spolnimi strukturami opredeljenih oseb. Za ostale dejavnosti so ciljne vrednosti izračunane kot povprečje na zdravnika.

Za posameznega izvajalca se ciljna vrednost določi kot seštevek ciljnih vrednosti po posameznih zdravnikih, glede na njihove zaposlitve po zdravstvenih dejavnostih.

#### **11.2.1.1 Osnovna zdravstvena dejavnost**

Za zdravnike v osnovni zdravstveni dejavnosti preračun temelji na razmerjih dejanskih vrednostih za predpisana zdravila na recepte po posameznih starostnih skupinah opredeljenih oseb, ločeno po spolu. Upošteva se število opredeljenih oseb pri osebnih zdravnikih, moških in žensk, v posamezni starostni skupini in pripadajoče vrednosti receptov, ki so bile predpisane v teh starostnih skupinah.

Spodnja tabela prikazuje primer izračuna količnikov za osnovno zdravstveno dejavnost. Zaradi varovanja podatkov so številске vrednosti nekoliko prilagojene.

*Tabela 5: Primer Prikaz izračuna količnikov za osnovno zdravstveno dejavnost za Slovenijo na podlagi podatkov za določeno leto*

	število opredeljenih	vrednost receptov	vrednost na opredeljenega	količniki
MOŠKI OD 0 DO 6	59.404	769.442.570	12.953	1,24
ŽENSKÉ OD 0 DO 6	55.792	665.903.019	11.935	1,14
MOŠKI OD 7 DO 19	143.230	859.840.636	6.003	0,58
ŽENSKÉ OD 7 DO 19	136.205	940.089.573	6.902	0,66
MOŠKI OD 20 DO 45	374.168	3.902.062.815	<b>10.429</b>	<b>1,00</b>
ŽENSKÉ OD 20 DO 45	367.482	4.873.339.755	13.261	1,27
MOŠKI OD 46 DO 64	241.065	11.126.978.185	46.158	4,43
ŽENSKÉ OD 46 DO 64	241.447	11.760.648.108	48.709	4,67
MOŠKI OD 65 DO 84	109.203	10.808.138.131	98.973	9,49
ŽENSKÉ OD 65 DO 84	171.285	17.485.518.664	102.084	9,79
MOŠKI 85 IN VEČ	4.750	462.471.198	97.362	9,34
ŽENSKÉ 85 IN VEČ	15.519	1.400.641.987	90.253	8,65
Skupaj	1.919.550	65.055.074.641		

Vir: ZZZS. (2007), *Predpisovanje zdravil na recepte po izvajalcih po metodologiji STIRA*, str. 3, Tabela 1.

Določi se količnik »1,00« kot povprečna vrednost zdravil na opredeljeno moško osebo v starosti od 20 do 44 let, koeficienti za vse ostale starostne skupine po spolu pa so opredeljeni v razmerju do te določene enote oz. vrednosti. S temi količniki se množi število opredeljenih oseb po posameznih razredih in izračuna število pogojno opredeljenih oseb za Slovenijo, za posameznega izvajalca in tudi za posameznega zdravnika.

Za posameznega zdravnika se število njegovih pogojno opredeljenih oseb množi z vrednostjo pogojne enote in dobimo ciljno vrednost za posameznega zdravnika (ZZZS, 2008).

### 11.2.1.2 Ginekologi z izbirami

Posebna skupina so tudi ginekologi, ki imajo v tej dejavnosti opredeljene osebe. Za izračun ciljne vrednosti za ginekologe z izbirami (tisti, ki imajo opredeljene osebe) so določene pogojne enote po vzoru iz osnovne zdravstvene dejavnosti, s tem da so za preračune upoštevani le recepti navedenih ginekologov in njihove opredeljene osebe.

Preračun temelji na razmerjih dejanskih zneskov za predpisana zdravila med posameznimi starostnimi skupinami. Upošteva se število opredeljenih v posamezni starostni skupini in koeficient dejanskega zneska za zdravila za vsako od teh starostnih skupin na podlagi realizacije stroškov.

Za koeficient z vrednostjo 1,00 je določen starostni razred od 13 do 45 let. V razmerju do te vrednosti se preračunajo vrednosti drugih dveh starostnih skupin in določijo razmerja

količnikov. S temi količniki se določi število pogojnih enot opredeljenih pri vsakem ginekologu in nadalje pomnoži z vrednostjo pogojne enote 1,00 za ginekologe.

### **11.2.1.3 Specialisti skupine 1 - ambulantna dejavnost**

Kriteriji so določeni ločeno za:

- zdravnike, specialiste skupine 1, ki se ukvarjajo z zdravljenjem v ambulantni, kot povprečna vrednost predpisanih zdravil na recepte na zdravnika zaposlenega v posamezni specialnosti,
- zdravnike specialiste skupine 2, ki se ukvarjajo z diagnostiko, upravljanjem in vodenjem pa se kot kriterij upošteva 50 receptov namenjenih za lastno uporabo, teh 50 receptov se pomnoži s povprečno vrednostjo recepta v Sloveniji.

### **11.2.1.4 Specialisti skupine 2**

Za zdravnike zaposlene v dejavnostih kot so rentgenologija, ultrazvočna diagnostika, socialna medicina, patologija, sodna medicina in podobne, je merilo za določanje ciljnega zneska za zdravila na račun zavarovanja 50 receptov, ki jih ima zdravnik za lastno uporabo, ki se pomnoži s povprečno vrednostjo recepta. Isto merilo velja za zdravnike, ki so zaposleni v upravnih in administrativnih službah ali so vodstveni delavci, čeprav občasno delajo v ambulantah.

### **11.2.1.5 Zobozdravstvo in maksilofacialna kirurgija**

Za zobozdravstveno dejavnost je ciljna vrednost opredeljena na podlagi povprečne vrednosti za zdravila na recepte na zobozdravnika v in zobozdravnika specialista skupaj v Sloveniji. Za maksilofacialno kirurgijo se upošteva povprečna vrednost na zdravnika v tej dejavnosti.

## **11.2.2 Obdelava podatkov in možna odstopanja**

Obdelava podatkov poteka na IC Zavoda s programom, ki v svojih procedurah povezuje podatke iz baze izbira osebnega zdravnika (IOZ), baze receptov, baze obveznega zdravstvenega zavarovanja (OZZ) in baze podatkov izvajalcev (BPI). Zato je pravilnost obdelav v celoti odvisna od točnosti in pravilnosti podatkov v teh bazah. Izvajalci so dolžni redno ažurirati vse spremembe nastale v zvezi z zaposlitvami svojih zdravnikov.

Kljub navedenemu lahko v praksi pride do odstopanj glede na dejansko stanje in to predvsem zaradi naslednjih dejstev:

- nadomeščanja zdravnikov, ki niso nikjer evidentirana;
- več kot ena zaposlitev zdravnika;
- sprememba zaposlitve v opazovanem obdobju;
- zaposlitve oz. lokacije izvajalcev v dveh območjih;
- upokojeni zdravniki, ki delajo pogodbeno;
- ni več »avtomatičnega prenosa« izbranih oseb, ko se zdravnik upokoji.

Zaposlitve zdravnikov v BPI so dinamične, zato je težko pričakovati, da bodo izpisi iz obdobja v obdobje enaki. Opažamo namreč, da je vedno več zdravnikov, ki imajo v bazi BPI evidentiranih po več veljavnih zaposlitev hkrati, pri enem ali več izvajalcih. Nastaja problem



razvrstitve takih zdravnikov, saj lahko trenutno uvrstimo zdravnika le k enemu izvajalcu. Program za obdelavo podatkov obravnava vse zdravnike po istem principu. Zato se v primerih, ko zdravnik predpisuje recepte na različnih lokacijah (ambulantah), recepti seštejejo po šifri zdravnika in razporedijo na zadnjo veljavno zaposlitev.

Podatke, ki so na razpolago iz obstoječe obdelave, je potrebno vedno interpretirati z določenimi omejitvami. Razlogi za to so v dinamiki podatkov iz baz podatkov, ki jih uporabljamo v rednih obdelavah. V manjši meri je možno, da podatki niso točni, saj lahko pride do napake že pri vhodnih podatkih. Izhodni podatki iz baze podatkov izvajalcev in izbira osebnega zdravnika so t.i. »stanje na dan«, med tem ko se podatki iz baze receptov nanašajo na določeno obdobje.

V obdelavo so zajeti recepti z zdravili pozitivne in vmesne liste, seveda tisti, ki so bili pravilno izpolnjeni. Recepti, ki nimajo izpolnjenih vseh podatkov se nahajajo pod oznako nerazvrščeni recepti 99999. Oznake pri posameznih zdravnikih pomenijo: + mrtev, & upokojen, % zaposlen izven zdravstva, \$\$ zaposlen pogodbeno. Te oznake nas opozarjajo na poseben status zdravnika v bazi podatkov izvajalca. Posebej je dodana oznaka VZ, ki pomeni, da ima zdravnik več zaposlitev (dve ali več, lahko tudi pri istem izvajalcu) (ZZZS, 2008).

### **11.2.3 Število in vrednost receptov ter indeksi odstopanja od ciljnih vrednosti za recepte po območnih enotah v določenem letu.**

Tako se vsako leto skupno in kvartalno izdelajo sezname (po zdravnikih, izvajalcih), koliko je bilo skupno izdanih receptov prednostne in vmesne liste ter koliko jih ni bilo zajetih v obdelavo zaradi nepopolnih podatkov. Največkrat je vzrok nepravilnih podatkov nepravilna šifra zdravnika na receptih, vendar jih je včasih razvrstiti v ustrezno območno enoto po šifri lekarne v kateri je bil izdan recept.

Predpisovanje zdravil je v grobem povezano z zdravljenjem velikega števila oseb s kroničnimi boleznimi, ki potrebujejo redno in dolgotrajno terapijo z zdravili, predvsem v osnovni zdravstveni dejavnosti ter zdravljenjem hujših oblik bolezni manjšega obsega oseb, ki potrebujejo nova draga zdravila. Hitra uvedba novih dragih zdravil za zdravljenje aidsa, multiple skleroze, ter draga onkološka zdravila, hormoni in citostatiki bistveno prispevajo k porastu skupne vrednosti zdravil, a imajo manj stranskih učinkov in prispevajo k zdravljenju bolezni, ter omogočajo čim manj hospitalizacij oseb. Upošteva se tudi predvidena stopnja inflacije in planirana rast odhodkov za zdravila iz obveznega zavarovanja.

### **11.2.4 Proces obdelave in logistika podatkov STIRA**

Obdelava podatkov STIRA in informiranja Zavodovih izvajalcev je proces, katerega lastnik je Področje za razvoj zdravstvenega zavarovanja ( v nadaljevanju PZRZZ). Proces lahko razdelimo na dva dela, prvi je strateški in drugi operativni.

1. **Strateški del procesa** se prične z določanjem ciljnih vrednosti za tekoče leto. V ta namen se izvede na IC (informacijski center) t.i. osnovna obdelava na podlagi podatkov preteklega leta (zagon PL programa) na podlagi katere nastanejo povprečne vrednosti receptov po posameznih specialnostih in količniki za osnovno dejavnost in ginekologije. Ta obdelava se posreduje na PZRZZ.

2. Iz teh podatkov PZRZZ pripravi predlog ciljnih vrednosti za tekoče leto z upoštevanjem ustrezne pričakovane inflacije in planirane rasti odhodkov za zdravila. Predlog ciljnih vrednosti potrdi direkcijski kolegij.
3. S temi aktivnostmi se strateški del procesa konča in prične **operativni del**, v katerem se potrjeni kriteriji in ciljne vrednosti pošljejo nazaj na IC, kjer se vključijo v drugo obdelavo podatkov (izvede programer na IC-u Zavoda).
4. Rezultat te obdelave so tri datoteke. Prva datoteka zajema število in vrednost receptov P in V liste, opredeljene osebe in število pogojno opredeljenih oseb ter ciljne vrednosti in indekse odstopanja za vsakega izvajalca, druga datoteka za tri krovne izvajalce in tretja datoteka za zdravnike.
5. V tej fazi se podatki iz teh datotek **kontrolirajo**. Kontrolo opravijo skrbnica baze receptov in predstavnica PZRZZ. Če je ugotovljeno, da podatki niso ustrezni, se obdelava (zagon PL programa) ponovi in ponovno kontrolira. V kolikor so bili podatki ustrezni, so se lahko iz teh datotek tvorili izpisi na papir. Ta del procesa je izvajal operativni oddelek IC. Izpisi so se nato v duplikatu dostavili na področje za razvoj, kjer jih je bilo potrebno razdeliti za posamezne območne enote, kuvertirati in kopirati zbirne tabele in poslati na Območne enote Zavoda za nadaljnje informiranje izvajalcev. Iz treh datotek je PZRZZ pripravil zbirne tabele v excelu, ki so se prav tako pošljale na Območne enote po elektronski pošti, vendar zaradi predhodno definirane tekstovne datoteke niso imele nobenih sumarnih vrstic.
6. Tak proces se izvede štiri krat letno in sicer na vsake tri mesece s kumulativni obdelavami podatkov.

Na podlagi testnih reportov Microstrategy se je ugotovilo, da bi bilo mogoče spremljanje podatkov oz. informiranje s podatki v delu, ki se nanaša na celotno logistiko in transformiranje tekstovnih datotek z reporti iz Microstrategy. Osnovna obdelava STIRA bi bila še vedno v pristojnosti programerjev PL datoteko pa bi posredovali informatiku z orodjem Microstrategy, ki bi iz nje izdelal v naprej definirana poročila.

Vzpostaviti se mora še vmesni člen med programiranjem in Microstrategy kot »nadzorni oz. kontrolni mehanizem«, kar bi v praksi pomenilo kontrolo in avtorizacijo izhodnih podatkov datoteke, preden bi se iz njih izdelala poročila.

Namen reportov Microstrategy za STIRA redne obdelave podatkov je v racionalizaciji aktivnosti pošiljanja izpisov in transformiranja tekstovnih datotek v excelove preglednice, pri čemer je prišlo do izgube sumarnih vrstic, glave tabele in drugih podatkov, ki so v tekstovni datoteki sicer zajeti, vendar zaradi količine neprimerni za excelove tabele (ZZZS, 2008).

### **11.2.5 Primerjava obstoječega in novega načina posredovanja podatkov STIRA**

Iz primerjave obstoječega in novega načina spremljanja podatkov in posredovanja podatkov izhaja, da dosedanji način kaže na določena dejstva z manjšo uporabno vrednostjo:

1. tekstovna datoteka nima sumarnih vrstic, zato jih tudi v excelovih ni in je potrebno ročno seštevanje, kjer je možnost napak večja;
2. velika količina papirnatih izpisov, velik obseg tiskanja na papir;

3. zaradi velike količine papirja so poštna pošiljke težke in drage;
4. zaradi tiskanja, dostave in logistike razpošiljanja je časovni zamik pri pridobivanju podatkov za območne enote daljši.

Hkrati pa novi način posredovanja podatkov z Microstrategy kaže določene dodane vrednosti za uporabnike, in sicer:

1. Hitro in enostavno pridobivanje podatkov in informacij za območne enote in za izvajalce,
2. Avtomatično obveščanje območnih enot o novih poročilih,
3. Reporti so elektronska oblika izpisov z vsemi podatki, omogočajo izračune in izpise sumarnih vrstic (totalov) in tudi drugih statističnih kazalcev,
4. reporti omogočajo tudi analitično spremljanje podatkov in statistične analize, katerih bi se jih lahko poslužile območne enote,
5. Izpis na papir samo po potrebi, v uporabniku bolj prijaznem formatu in obliki,
6. Manjša možnost napak pri analiziranju podatkov.
7. Orodje Microstrategy ponuja še veliko možnosti za raziskovanje po podatkih ter s tem pridobivanja novih koristnih informacij, ki pripomorejo k hitrejšemu odločanju (ZZZS, 2008).

### **11.3 Načrtovanje podatkovnega skladišča za projekt STIRA**

Na kratko je podatkovno skladišče baza, ki je namenjena branju podatkov in služi analitičnim postopkom pri podpori odločanju. Sliši se dokaj enostavno, a je od začetne do končne faze precej dolga in naporna pot. Ni pomembno ali je to celovito podatkovno skladišče ali posamezna podatkovna tržnica, pri vsakem se je potrebno načrtovanja lotiti s premislekom in kar je najbolj pomembno, da v začetni fazi določimo, kakšen bo namen in cilj podatkovnega skladišča.

Pristop k izgradnji skladišča podatkov:

- definirati organizacijo dela: določiti skrbništvo sistema ter izvajanje nalog;
- definirati potrebne korake za vzpostavitev sistema;
- določiti arhitekturo sistema;
- analiza in zbiranje zahtev uporabnikov;
- načrtovanje dimenzijskega modela ter na podlagi tega izgradnja podatkovne baze;
- vsebina in izvedba procesa polnjenja, čiščenja in transformacije podatkovnega skladišča;
- načrt za polnjenje baze iz operativnih podatkov;
- implementacija rešitve v orodju olap;
- testiranje, vzdrževanje in dokumentiranje rešitve.

#### **11.3.1 Organizacija dela**

Organizacija dela pri je razdeljena na administratorje različnih področij orodja Microstrategy ter izvajanje nalog od začetne do končne faze izgradnje podatkovnega skladišča.

##### **1. Skrbništvo sistema**

Skrbnišтво sistema je razdeljeno na administratorje različnih področij. Podrobnosti prikazuje naslednja preglednica.

*Tabela 6: Skrbništvo sistema orodja Microstrategy*

Vloga	Nosilec
<b>Administrator strojne opreme in operacijskega sistema strežnikov</b> (razvojno-testnega) in produkcijskega	Oddelek za mreže, kjer zadolžene imenuje vodja oddelka
<b>Administrator Microstrategy Intelligence Server</b> – skrb za instalacije, za nadzor, za postavitve in vzdrževanje pooblastil, za standarde v zvezi s tem produktom, za vzpostavitev backup procedur, na vzpostavitev nadzornih rešitev, za dvigovanje verzij	Nosilec: oseba s področja administracije podatkovnih baz, ki ima tudi znanje iz področja administracija Microstrategy Intelligent Serverja.
<b>Administrator orodja Microstrategy Architect</b> – definiranje razvojnih standardov, nastavitvev, postopkov	Dve osebi iz ekipe s področja podatkovnih skladišč.
<b>Administrator orodja Microstrategy Desktop</b> – definiranje razvojnih standardov, nastavitvev	Dve osebi iz ekipe s področja podatkovnih skladišč.

Vir: ZZZS. (2008),, Dokumentacija projekta STIRA, str. 2.

## 2. Izvajanje nalog

V projekt STIRA je bilo vključenih kar nekaj sodelujočih. Vloge pri izvajanju nalog prikazuje spodnja preglednica.

*Tabela 7: Vloge pri izvajanju nalog za projekt STIRA*

Vloga	Nosilec
<b>Vodja skrbniške ekipe</b> , ki je obenem tudi <b>projektant</b> – njegova vloga je usklajevanje naročil uporabnikov, planiranje in organiziranje izvedbe, dogovarjanje s skrbniki aplikacij za zagotovitev vhodnih podatkov, načrtovanje rešitev, testiranje rešitev, priprava navodil za uporabnike, uvedba rešitev, pomoč uporabnikom.	Sodelovanje treh iz ekipe za podatkovna skladišča. Mene kot razvijalca, vodje skupine ter osebe, ki je kot pomoč pri projektiranju in razvoju. Pri povezovanju podatkov različnih poslovnih področij pomaga uporabnik za načrtovanje podatkovnih in procesnih modelov.
<b>Razvoj postopkov za pripravo vhodnih podatkov, priprava podatkov.</b>	Izvajajo skrbniške ekipe tistih aplikacij, iz katerih se pridobivajo podatki.
<b>Razvijalec rešitev</b> – uvažanje podatkov v orodje Microstrategy, priprava kock za analiziranje podatkov, ureditev pooblastil, predpriprava poročil, pomoč uporabnikom pri pripravi poročil (kateri ne bodo znali sami).	V ekipi za razvoj podatkovnih skladišč sva dva razvijalca rešitev.

Vir: ZZZS. (2008), Dokumentacija projekta STIRA, str. 2.

### **11.3.2. Potrebni koraki za vzpostavitev sistema**

Za vzpostavitev sistema za delovanje podatkovnega skladišča je bilo potrebno zagotoviti kar nekaj korakov. Plan nam je v začetku koristil pri planiranju nalog tudi za zunanjega izvajalca.

#### **1. Izobraževanje notranje ekipe**

Potrebno je bilo pridobiti potrebna temeljna znanja za obvladovanje vlog na projektu:

- modeliranje in načrtovanje podatkovnih kock: Nekateri v ekipi so že imeli nekaj osnovnega znanja, ostali so se morali z omenjenega področja še izobraziti. Za pridobivanje znanja smo se obračali tudi na strokovno literaturo ter primere iz prakse (obstoječe podatkovno skladišče DIS – direktorski informacijski sistem). Dodeljena nam je bila tudi poskusna naloga načrtovanje podatkovne tržnice za primer analiziranja podatkov za skrbniške naloge od katere smo se vsi zelo veliko naučili. Sama sem se v novembru 2007 v Amsterdamu udeležila tudi tečaja Dimenzijsko modeliranje v globino (Dimenzional Modelling in Depth), ki ga je vodil Ralph Kimball ter Margy Ross, strokovnjaka iz omenjenega področja. Tečaj mi je zelo poglobil zanje s področja dimenzijskega modeliranja.
- obvladovanje Microstrategy tehnologije – v novembru 2006 se je celotna ekipa udeležila izobraževanja uporabe orodja Microstrategy.

#### **2. Ureditev strojne in systemske programske opreme**

Zagotoviti je bilo predvsem potrebno razvojno-testni in produkcijski strežnik. Uporabniki, ki bodo uporabljali orodje Microstrategy, bodo do uporabe poročil dostopali preko spleta, tako da za delo potrebujejo Internet Explorer ali kak drug ustrezen spletni program. Razvijalca sva bila dva, kjer sva imela na osebнем računalniku nameščen Microstrategy Desktop, Intelligent Server (IS) bi bil potreben le v primeru razvijanja nadzornih plošč in naprednih Microstrategy dokumentov. Lahko pa se v omenjenem primeru poveže tudi na testni strežnik. Kot razvijalec, da je sistem deloval dokaj hitro sem morala imeti na računalniku vsaj 1,00 GB RAM-a, 994 MHz, imam računalnik Microsoft Windows XP Professional, Različico 2002, Service Pack 2.

#### **3. Ureditev instalacij Microstrategy orodij**

Potrebno je bilo preveriti ali imamo inštalirane vse produkte in ali so ustreznih verzij. Pripraviti je bilo potrebno tudi postopke za backup in nadzor. Pri definiranju uporabnikov in njihovih pravic je bilo potrebno določiti, katere dodatne licence potrebujemo. Omenim naj le, da je bilo večino uporabnikov definiranih kot Web professional userji, kar jim omogoča, da lahko na poročilih spreminjajo oblike ter odvezemajo ali dodajajo objekte (atributi, mere, filtri,...). Ostali so bili definirani kot Web Analyst uporabniki ter Web Reporterji. Funkcionalnosti definiranih uporabnikov so opisane v prilogi pod opisom orodja Microstrategy.

#### **4. Ureditev razvojnega okolja**

Potrebno je bilo naštudirati definiranje uporabnikov ter možnosti pooblastil. Omenjeno smo se naučili na tečaju za orodje Microstrategy. Definirati je bilo potrebno razvojno, testno in produkcijsko okolje in baze podatkov. Ureditev okolja dobro prikazuje slika arhitekture sistema, ki je opisana v nadaljevanju.

#### **5. Poskusni razvoj manjših rešitev**

Že zgoraj je bilo omenjeno, da je ekipa dobila testno nalogo, kjer je bil glavni namen predvsem učenje in nabiranje izkušenj.

## **6. Razvoj internih standardov**

Pripraviti čim več standardov, da bo razvoj kar se da enoten in kontroliran.

### **11.3.3 Opis arhitekture sistema**

Omenjeno je že bilo, da se zaradi manj uspešnega projekta v preteklosti, ko se je zgradilo celovito podatkovno skladišče Nadzornik, ekipa na področju podatkovnih skladišč ukvarja z izgradnjo manjših podatkovnih tržnic. Glavni namen je ad-hoc analitično poročanje. Podatkovne tržnice naj bi se načrtovale le z namenom za analiziranje trenutnih aktualnih problemov. S tem namenom je bilo načrtovano in zgrajeno tudi podatkovno skladišče STIRA. Kmalu se je izkazalo, da ne bo šlo le za enkratno polnjenje, temveč periodično, ki se bo izvajalo vsake tri mesece. Uporabniki so spoznali, da uporaba olap orodja Microstrategy prinaša dodatne koristi in nove možnosti raziskovanja po podatkih.

Poznamo dva načina pristopa do meta podatkov, in sicer:

- direktni, kjer se do meta podatkov povežemo preko ODBC-ja in
- server pristop, kjer uporabimo meta bazo inteligent serverja.

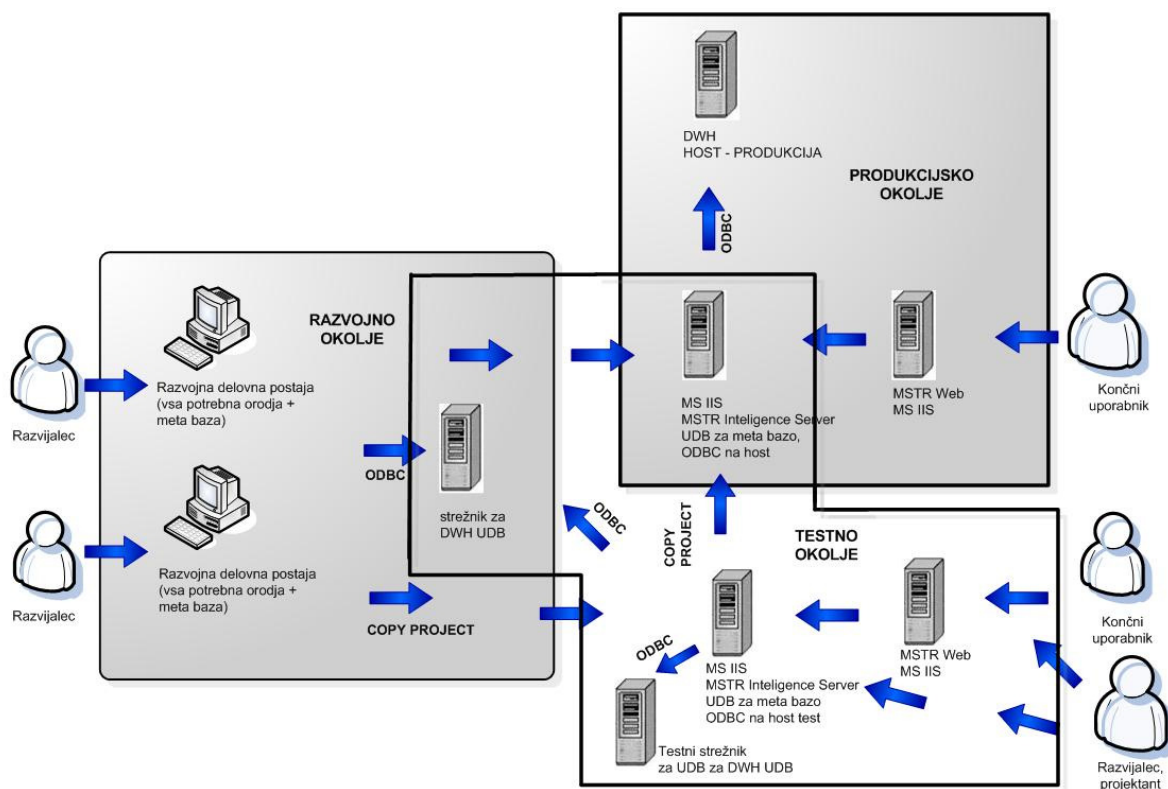
Arhitekturo sistema smo načrtovali natančno, saj smo morali upoštevati tudi varnostne karakteristike sistema. Arhitektura sistema je deloma tudi odvisna od kupljenih licenc olap orodja v našem primeru Microstrategy-ija.

Razvojna postaja ima naloženo orodje Microstrategy Desktop, Inteligent Server ni potreben, razen v primeru razvoja dinamičnih razvijnih plošč ali naprednih dokumentov. Ker smo omenjena področja le testirali v namene osebnega izobraževanja, smo imeli v tem času naložen le testni IS za določeno obdobje. Strežnika za razvoj in produkcijo, kjer je naložen Microstrategy Intelligent Server, sta ločena. Na omenjenih strežnikih mora biti vse, kar strežnik potrebuje vključno z ODBC povezavami. V okviru enega inteligent serverja imamo lahko samo eno lokalno metabazo.

Za testno-razvojno okolje se dostop do podatkovno skladišče nahaja lokalni DB2 UDB bazi, meta baza je kreirana v Access-u. Podatkovno skladišče za produkcijo, se nahaja na centralnem računalniku, saj na Zavodu že obstaja tam produkcijsko okolje za podatkovno skladišče. Postavitev podatkovne tržnice STIRA na centralnem računalniku hkrati predstavlja bolj varno, stabilno in zaščiteno okolje, kot na lokalnem strežniku. Končni uporabniki dostopajo do uporabe poročil orodja Microstrategy preko spletnega brskalnika. Strežnika za test in produkcijo, kjer je naložen Microstrategy WEB sta ločena. Tam so naloženi tudi vsi potrebni servisi za zagon. Za njihovo delovanje skrbi administrator, ki je zadolžen tudi za delovanje Inteligent Serverja.

Postavitev arhitekture sistema prikazuje Slika 10.

Slika 10: Arhitektura sistema projekta STIRA



Vir: ZZS. (2008), Povzetek arhitekture rešitve, str. 5, slika 3.

### 11.3.4 Analiza in zbiranje zahtev uporabnikov

Z vsebino in delovanjem področja STIRA smo se seznanili predvsem s pomočjo krajših delovnih sestankov, kjer so bili prisotni naslednji udeleženci:

- vodja sektorja za razvoj od katerega smo prejeli glavne smernice za razvoj podatkovnih skladišč, izvedeli kaj je okvirna vsebina projekta STIRA ter kateri ljudje so vključeni v proces;
- PL programer: izvedeli smo vsebino PL programov, ki so v starem sistemu omogočali izpis ključnih podatkov na papir;
- začetni ključni uporabniki s področja PZRZZ ter skrbnica baze receptov: opravijo kontrolo podatkov v datotekah. Od omenjenih uporabnikov smo se zelo veliko naučili, saj dobro poznajo vsebino projekta STIRA. Kar pa je najbolj pomembno, so tudi dokaj dobro informacijsko podkovani, kar nam je bilo v dodatno pomoč. Na začetku smo sodelovali le s tremi ključnimi uporabniki, končno zaključno izobraževanje je bilo za večje število predstavnikov območnih enot.
- predstavnik operativnega oddelka: v operativnem oddelku so se izpisi tiskali na papir
- naša ekipa: administrator baz podatkov ter Intelligent Serverja, razvijalci in načrtovalci podatkovnih skladišč ter vodja naše skupine.

V primeru raznih nejasnosti smo vprašanja reševali tudi po telefonu ali elektronski pošti. Pred zbiranjem zahtev je pomembno, da smo se najprej seznanili o vsebini projekta STIRA ter šele nato določili zahteve uporabnikov oziroma cilje novega sistema. Pri tem smo poiskali tudi

določene nazive tudi drugje v bazi. Omenjeno je že bili, da smo za ključni uporabniki izdelali najprej testno verzijo, kjer se sta se lahko seznanili tudi z samim orodjem Microstrategy ter nam tako lažje svetovali, kakšna poročila potrebujeta.

Od uporabnikov smo izvedeli, da bodo glavni izvorni podatki že delno agregirane ASCII datoteke PL programerja, ki jih bomo prejeli ob zaključku vsakega kvartala. Podatki v datotekah so zbrani sumarno, torej za prvi kvartal, od prvega do drugega kvartala, od prvega do tretjega ter od prvega do četrtega kvartala.

Začetne zahteve uporabnikov so bile, da bi poročila vsebovala sledeče podatke:

- Izvajalec:
  - Območno enoto (v nadaljevanju OE), Ključ izvajalca, Naziv izvajalca;
- Dejavnost: Dejavnosti v katero spada določen izvajalec (npr. osnovna dejavnost, zobozdravstvena dejavnost);
- Zdravnika:
  - Ključ zdravnika, Naziv zdravnika.
- Oznako zdravnika: oznaka zdravnika pomeni njegov status. Oznake pri posameznih zdravnikih pomenijo: + mrtev, & upokojen, % zaposlen izven zdravstva, \$\$ zaposlen pogodbeno;
- Merljive vrednosti:
  - Število receptov, Število opredeljenih oseb, Število pogojnih opredeljenih oseb, Vrednost receptov, Ciljna vrednost ter Indeks.

Želeli so imeti zajete omenjene podatke po izvajalcih in zdravnikih glede na dejavnost ter samo za izvajalce glede na dejavnost. Poročila naj bi vsebinsko izgledala tako, kot prikazuje primer spodaj.

*Tabela 8: Struktura poročil za projekt STIRA – prikaz podatkov po izvajalcih in zdravnikih glede na dejavnost.*

Izvajalec				Zdravnik		Mere	Št. rec.	Št. opred. oseb	Št. pog. opred. os	Vred. rec.	Ciljna vred.	Indeks
OE	Klj izvajalca	Naziv izv.	Dej.	Klj zdr.	Naziv zdr.	Oznaka zdr.						

Vir: ZZS. (2008), Dokumentacija projekta STIRA, str. 1

*Tabela 9: Struktura poročil za projekt STIRA – prikaz podatkov po izvajalcih glede na dejavnost.*

Izvajalec				Mere	Št. rec.	Št. opred. oseb	Št. pog. opred. os	Vred. rec.	Ciljna vred.	Indeks
OE	Klj izvajalca	Naziv izv.	Dej.	Oznaka zdr.						

Vir: ZZS. (2008), Dokumentacija projekta STIRA, str. 1



### 11.3.5 Načrtovanje dimenzijskega modela

Veliko glede tega je bilo že omenjeno v prejšnjih poglavjih. Poznamo dva načina pristopa gradnje podatkovnega skladišča, in sicer od zgoraj navzdol ter od spodaj navzgor. Pri projektu STIRA pa smo uporabili kombiniran pristop, saj se nam je zdel glede na vsebino in zahteve uporabnikov najprimernejši. Od uporabnikov smo prejeli zahteve, kaj želijo od podatkovnega skladišča, nato smo poskusili poiskati določene vire tudi drugje v bazi. To so bili predvsem razni nazivi za polnjenje dimenzijskih tabel. Nato smo izdelali prototipni dimenzijski model, razvili prototipno rešitev v orodju Microstrategy ter poučili uporabnike glede uporabe orodja. Tako so je pojavili določene zahteve po dopolnitvah in določenih spremembah dimenzijskega modela. Tako smo ponavljali toliko časa, da so uporabniki potrdili rešitev.

Glede na njihove zahteve smo hitro lahko izluščili elemente dimenzijskega modela. Že ko se je načrtoval dimenzijski model, smo morali imeti v predstavah, kako naj izgledajo testna poročila v orodju Microstrategy. Iz uporabniških zahtev, kaj želijo spremljati in po čem želijo spremljati prikazane podatke smo določili dimenzije in mere.

Po Kimballu smo določili:

- poslovni proces: spremljanje receptov po zdravnikih in izvajalcih glede na dejavnost;
- poslovno podrobnost (angl. *Grain*): to je ena vrstica v tabeli dejstev: Območno enoto Izvajalca, Ključ izvajalca, Naziv izvajalca, Dejavnost, Ključ zdravnika, Naziv zdravnik, Oznako zdravnika, Število receptov, Število opredeljenih oseb, Število pogojnih opredeljenih oseb, Vrednost receptov, Ciljna vrednost ter Indeks.
- dimenzije ali opisne vsebine: Območno enoto Izvajalca, Ključ izvajalca, Naziv izvajalca, Dejavnost, Ključ zdravnika, Naziv zdravnik, Oznako zdravnika
- numerične mere: Število receptov, Število opredeljenih oseb, Število pogojnih opredeljenih oseb, Vrednost receptov, Ciljna vrednost ter Indeks.

Glede na zgoraj opisano ter po manjših prilagajanjih smo oblikovali dimenzijski model, ki je prikazan na sliki 11. Zaradi lažje preglednosti dimenzijskega modela dimenzije časa ni v celoti narisana v dimenzijskem modelu. Tabele se povezujejo preko tujih ključev, tako da je tabela leto povezana s tabelo kvartal, tabela kvartal s tabelo mesec, tabela mesec s tabelo dan. Tabela dan ima torej tuje ključne vse naštetih.

Definirane se bile tabele, ki jih vključuje podatkovno skladišče STIRA:

- tabela zdravnikov
- tabela izvajalcev
- tabela dejavnosti
- tabela oznak zdravnika
- fact tabela ali tabela dejstev (vrednosti receptov, število receptov, indeksi, itd.)
- agregatna tabela določenih krovnih izvajalcev: Krovni izvajalci so: KC Ljubljana, zdravstveni dom Ljubljana, Terme Krka, ZG Kranj. Agregatna tabela se za nazive ravno tako povezuje z ustreznimi dimenzijskimi tabelami, čeprav tega na dimenzijskem modelu zaradi preglednosti ni prikazano.

Opisi tabel so priloženi v prilogi.

Slika 11: Dimenzijski model za projekt STIRA



Vir: ZZS. (2008), *Povzetek arhitekture rešitve*, str. 2, slika 3.

### 11.3.6 Vsebina procesa polnjenja podatkovnega skladišča

S formalizacijo aplikacije STIRA smo pridobili možnost za upravičenost do začasnih pooblastil na produkcijskih podatkih. Vsebinski skrbnik vsak konec kvartala poda novo zahtevo za polnjenje podatkovnega skladišča, odpre se nova skrbniška naloga, dodelijo se pooblastila do produkcijskih podatkov. To je za pravice do produkcijskih podatkov za nazive zdravnikov in izvajalcev.

Postopek ETL-a oziroma zbiranje, transformacija, čiščenje in polnjenje podatkov v podatkovno skladišče ponavadi zahteva največ časa v procesu izgradnje podatkovnega skladišča. Čas je velikokrat odvisen tudi od kvalitete vhodnih podatkov. Dobro in kvalitetno zgrajeno podatkovno skladišče je torej pogoj za izdelavo pravih analiz in zato zahteva

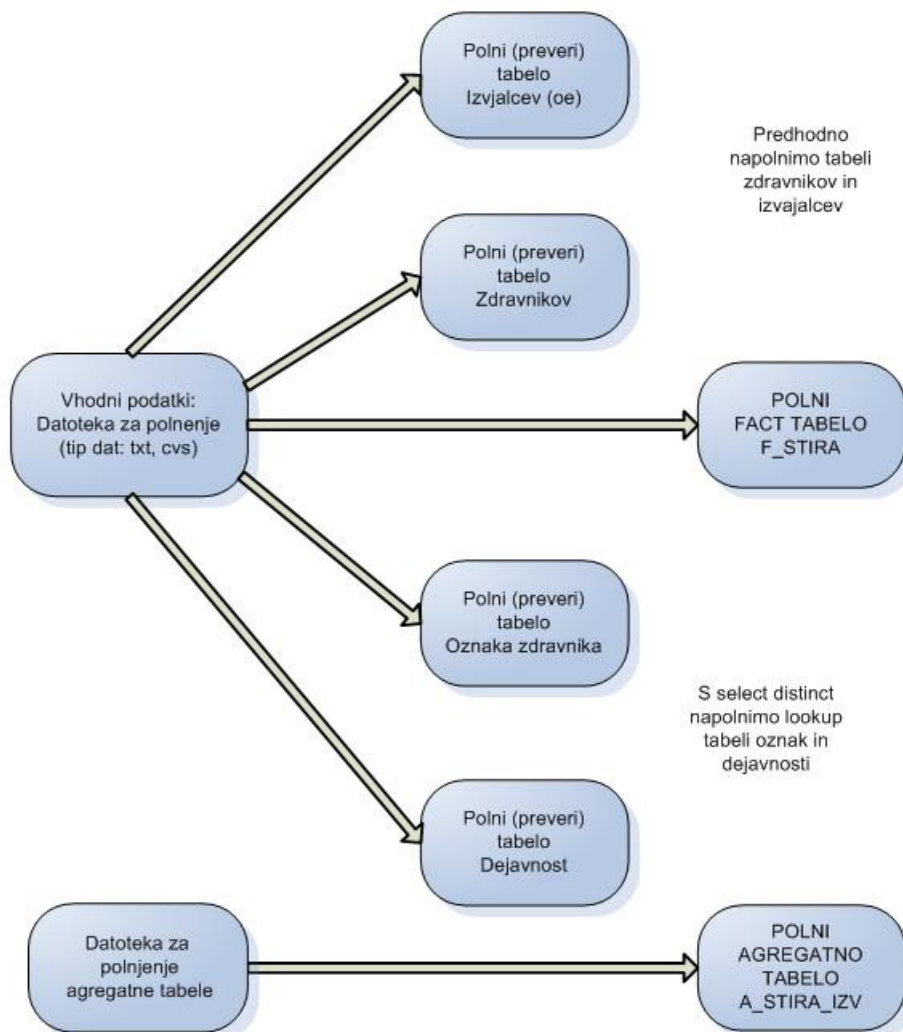
posebno pozornost. Skrbnica podatkov je na enem izmed sestankov zagotovila, da kvaliteta podatkov glede na količino podatkov (8 mio receptov, 6-8000 zdravnikov,...) precej dobra. Predvsem se je izboljšala na račun podprojekta za čiščenje kvalitete podatkov. Šlo je za popravke programske opreme v lekarnah. Podatki se še izboljšujejo, nobena tabela pa ni 100% pravilna.

Že v prejšnjem poglavju je bilo omenjeno, da se podatkovno skladišče polni kvartalno. Prejmemo že delno agregirane podatke v obliki treh ASCII datotek: po zdravnikih, samo po izvajalcih ter krovni izvajalci. Podatki v datotekah so zbrani sumarno. S pomočjo te datoteke ter nekaterih šifrantov (najprej iz testnega nato produkcijskega) se nato napolni dimenzijske tabele v podatkovnem skladišču.

Za polnjenje nazivov za tabeli zdravnikov in izvajalcev so se uporabljali podatki iz testnega okolja ter kasneje iz produkcijskega.

Spodnja shema grafično prikazuje način polnjenja podatkovnega skladišča.

Slika 12: Prikaz načina polnjenja podatkovnega skladišča



Vir: ZZZS. (2008), *Povzetek arhitekture rešitve*, str. 7, slika 5.

Iz prenosne datoteke smo najprej napolnili dimenzije podatkovnega skladišča. Dimenzije smo vedno sproti preverjali in vnesli manjkajoče podatke. Vedno smo se držali pravila, da v tabelah dimenzijskega modela ne bomo imeli podatkov z vrednostjo null. To nam kasneje olajša delo z orodjem Microstrategy. Tako smo za določeno šifro zdravnika, kjer ni bilo podanega naziva, vnesli besedi: neznan in nedoločen. Prav tako smo vnesli, če ni bilo dejavnosti, naziv Nedoločena dejavnost in prazno vrednost, če ni bilo oznake zdravnika.

Potek smo izvajali v naslednjem vrstnem redu:

- naprej smo preverili in polnili vse dimenzije: tabele izvajalcev, zdravnikov, oznak in dejavnosti;
- ko smo imeli vse pravilne dimenzije, smo polnili tabelo dejstev. V tabeli dejstev ne more biti zapisa, ki ne bi ustrezal dimenzijam. Lahko pa dimenzije vključujejo zapise, ki jih ni v tabeli dejstev;
- polnili še agregatno tabelo: omenjena tabela vključuje podatke za krovne izvajalce.

SQL-i za polnjenje in preverjanje so podani v prilogi.

### 11.3.7 Izbira tehnologije oziroma načina polnjenja podatkovnega skladišča

ETL orodja so vrsta orodij, ker se ni več potrebno spraševati, ali bo mogoče dobiti podatke, ali lahko ugotovimo izvor podatkov, ali bomo dovolj hitro napolnili podatkovno skladišče, kako bomo implementirali kvaliteto podatkov ipd. Na tržišču dosegajo ETL orodja dokaj visoko ceno. ETL orodja niso enostavna za uporabo, zato se je za njihovo uporabo potrebno ustrezno izobraziti (NPS, 2008).

Postopek zajemanja, transformacije in nalaganja podatkov v podatkovno skladišče ali na kratko ETL proces nam je vzel kar veliko časa za premislek. Zavedali smo se, da je omenjeni postopek eden izmed najpomembnejših v razvoju podatkovnega skladišča. Iskali smo rešitve, da bi proces polnjenja, kar se da najbolj avtomatizirali. Pravilnost prenosa podatkov je pomembna za točne analize nad podatki v podatkovnem skladišču.

Odločali smo se med naslednjimi opcijami, ali kupiti ETL orodje, ali razviti proceduro za polnjenje. To so lahko Java Store Procedure ali ročno napisane procedure v poizvedovalnem jeziku.

Izvedli smo testiranje prenosa podatkov na dva načina, in sicer:

- s pomočjo ETL orodij: DWE – ki je komponenta DB2 UDB baze, DataStage-a (podjetja IBM) in Informatica (podjetje Informatica).
  - Omenjena orodja nam omogočajo:
    - avtomatiziran proces polnjenja dimenzijskih tabel in fact tabele;
    - enostavno določitev urnika izvajanja;
    - istočasno povezljivost več okolij naenkrat (lokalna baza, baza na oddaljeni lokaciji, povezava na mainframe,...);
    - močno podporo za uporabo raznih funkcij, ki jih lahko koristno uporabimo za čiščenje podatkov, itd.;
    - dobro dokumentacijo postopkov (s tem se olajša delo delavcem, ki bi delo opravljali za nami).
- s pomočjo ročnega polnjenja za kreiranje tabel ali vnosa podatkov. Pri ročnem polnjenju smo se srečevali s problemi čiščenja podatkov (npr. decimalne pike).

ETL orodja so naložba za prihodnost, so zmogljiva in nam omogočajo zelo dobro dokumentacijo celotnega procesa polnjenja. Bilo je kar precej pomislekov glede nakupa ETL orodja, saj se na trgu razmeroma draga. V letošnjem letu so se na Zavodu odločili za nakup orodja DataStage. Glede na testiranja se orodje izkazalo kot najbolj uporabno za polnjenje podatkovne tržnice. Prednosti in kratek opis polnjenja ter uporabe je na kratko opisano v naslednjih poglavjih.

Pomanjkljivosti orodja DWE proti DataStage-u in Informatici:

- za uporabnika nerodno definiranje ponorov in izvorov podatkov;
- vse spremenljivke in sql stavki se kličejo preko predhodno določenih txt ali sql datotek;
- je striktno vezan na bazo DB2.

Informatica je glede na vse funkcionalnosti in zmogljivosti veliko bolj obsežna kot DWE in DataStage. Za obvladovanje orodja bi potrebovali precej več časa. Omogoča: integracijo podatkov, sinhronizacijo, verzioniranje, podpira SOAP tehnologijo (web services), enoten pogled na podatke, migracije podatkov in še mnogo drugega.

Kjer lahko že pred uporabo orodij ETL (ki večkrat niso sposobna obvladovati široke množice problemov), uporabimo posebna orodja za prečiščevanje. Tako za polnjenje različnih struktur uporabimo prečiščene, konsistentne podatke iz področij priprave. Uporaba orodij za prečiščevanje in orodij za povratni inženiring pred uporabo orodij ETL zmanjšuje probleme s kakovostjo podatkov v podatkovnem skladišču in lahko v veliki meri pripomore k večjemu uspehu projekta podatkovnega skladišča in k hitrejši povrnitvi vloženi sredstev (Golob, Welzer, 2000).

### **11.3.8 Polnjenje podatkovnega skladišča s pomočjo orodja DataStage**

Orodje DataStage se je glede na testiranje, ki smo ga izvedli izkazalo kot najbolj primerno za polnjenje podatkovnega skladišča. Glede na ostala dva orodja je bolj prijazen za uporabo, tudi čas učenja je proti orodju Informatica krajši. Tudi IBM-ov predstavnik, ki nam je orodje predstavil nam je priporočil omenjeno orodje glede na naše okolje in zahteve (SW: ZOS DB2, WIN DB2, Data Stage, MicroStrategy).

Kratek opis orodja:

- DataStage je napredno orodje, ki omogoča na hiter in enostaven način pripraviti podatke za podatkovna skladišča, področna podatkovna skladišča, ali za migracijo podatkov med različnimi sistemi v podjetju;
- nudi podporo široke palete različnih virov za izvor in ponor;
- procesiranje in transformacija velikih količin podatkov;
- vgrajena podpora za enostavne in zapletene transformacije podatkov;
- omogoča direktno (native) povezavo do različnih virov in aplikacij kot izvor ali ponor;
- orodje nam omogoča tudi dobro podporo pri odkrivanju napak;
- pregledno definiranje procesov izvajanja ter njihovo povezavo v logično zaporedje;
- natančno spremljanje poteka izvajanja posla;
- omogoča dobro dokumentacijo procesov.

Programsko orodje DataStage je sestavljeno iz štirih samostojnih orodij oziroma odjemalcev, ki se med seboj povezujejo (Tabela 10).

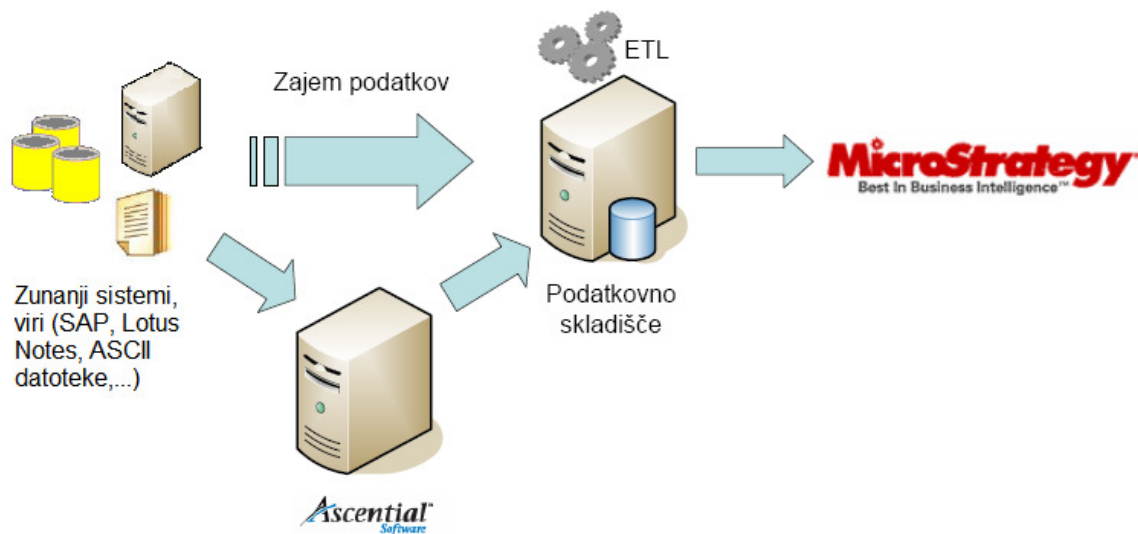
Tabela 10: DataStage odjemalci

<b>DataStage Administrator:</b> omogoča administracijo projektov in nastavitve DS strežnika, projekta(ov)	<b>DataStage Director:</b> omogoča izvajanje in opazovanje izvajanja poslov
<b>DataStage Designer:</b> Omogoča kreiranje poslov (opravil), prevajanje poslov v izvršljive programe za DataStage Server	<b>DataStage Manager:</b> Omogoča vpogled in upravljanje z repositorijem

Vir: avtor.

Spodnja slika prikazuje umestitev orodja DataStage v sistem Zavoda.

Slika 13: Prikaz umestitve orodja DataStage ter povezava z Microstrategy



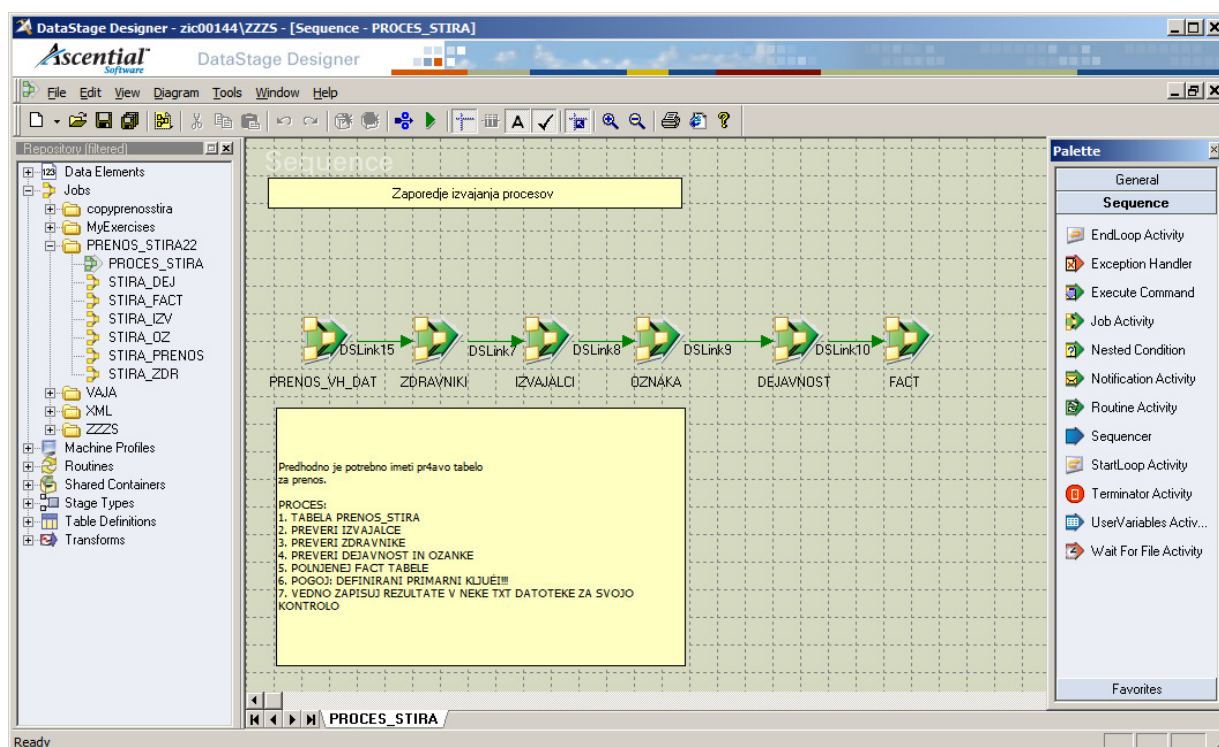
Vir: ZZZS, Dokumentacija projekta STIRA, 2008.

Glede na vsebino polnjenja za projekt STIRA so bili v DataStage Designerju definirani naslednji procesi:

- preverjanje in vnos novih zdravnikov (če obstajajo novi zdravniki, jih vnesemo);
- preverjanje in vnos novih izvajalcev (če obstajajo novi izvajalci, jih vnesemo);
- preverjanje in vnos novih dejavnosti (če obstajajo nove dejavnosti, jih vnesemo);
- preverjanje in vnos novih oznak zdravnika (enako preverimo oznake zdravnikov);
- polnjenje Fact tabele stira (ko imamo vse lookup tabele, napolnimo še fact tabelo).

Spodnja slika prikazuje zaporedje povezanih procesov za projekt STIRA. To je t.i kontrolni tok podatkov (angl. *control flow*). Pred tem definiramo vsak proces posebej, določimo izvor, ponor podatkov ter vmesne možne transformacije in agregacije podatkov. Nato omenjene podatkovne tokove povežemo v proces izvajanja, v DataStage imenovan tok zaporedja (angl. *Sequence job*) (Slika 14).

Slika 14: Sequence job oz. control flow (kontrolni tok) za povezovanje posameznih podatkovnih procesov med seboj.



Vir: Orodje DataStage Ascential - DataStage Designer, 2008

Omogočeno nam je izvajanje vsakega joba skupaj ali ločeno. Procese lahko avtomatiziramo ter določimo urnike izvajanj. Razvidno je bilo, da se za celoten potek polnjenja potrebuje natanko 2:51 minute, kar je razmeroma malo za prenos 27830 kB podatkov. Sam proces zaganjanja ročnih query-ijev bi bil s pomočjo Javastore procedur ali posameznim ročnim polnjenjem nekoliko hitrejši, a nam na dolgi rok zagotavlja dobro dokumentacijo ter hitrejše ponovno polnjenje. Hitro lahko s pomočjo dokumentacije ugotovimo, od kje smo dobili podatke, kako je potekalo polnjenje, kakšne transformacije so bile uporabljene, itd. Omogočeno nam je hitrejše obnavljanje procesov, kadar je potreba po nadgradnji podatkovnega skladišča. Glede na to, da je potrebno kvartalno polnjenje za področje STIRA ter da bo verjetno prihajalo do dodatnih zahtev končnih uporabnikov, bi lahko ETL orodje koristno uporabili. V našem primeru za področje STIRE, ni bilo potrebno kreirati zahtevnih transformacij, saj je vhodna datoteka že vsebovala dobro strukturirane in agregirane podatke. Pri nadaljnjih nalogah, s katerimi se bomo srečevali, pa nam bi omenjene prednosti (uporaba funkcij, rutin, transformacij, if else stavkov,...) koristile. Z razliko od orodja DWE nam DataStage istočasno omogoča povezavo z različnimi viri podatkov (testirali smo povezavo na mainframe, oddaljeno lokacijo DB2, uporabljali ASCII ter XML datoteke) ima tudi dobro podporo za uporabo XML-ja. Vseh možnosti in funkcionalnosti, kar nam jih orodje DataStage



ponuja, nismo uspeli stestirati. ETL orodja so glede na prednosti, ki jih imajo, dobra naložba za polnjenje podatkovnih skladišč ali tržnic.

Kratek opis omenjenega orodja je priložen v prilogi.

### 11.3.9 Implementacija analiz z orodjem Microstrategy ter kratka predstavitev uporabe orodja

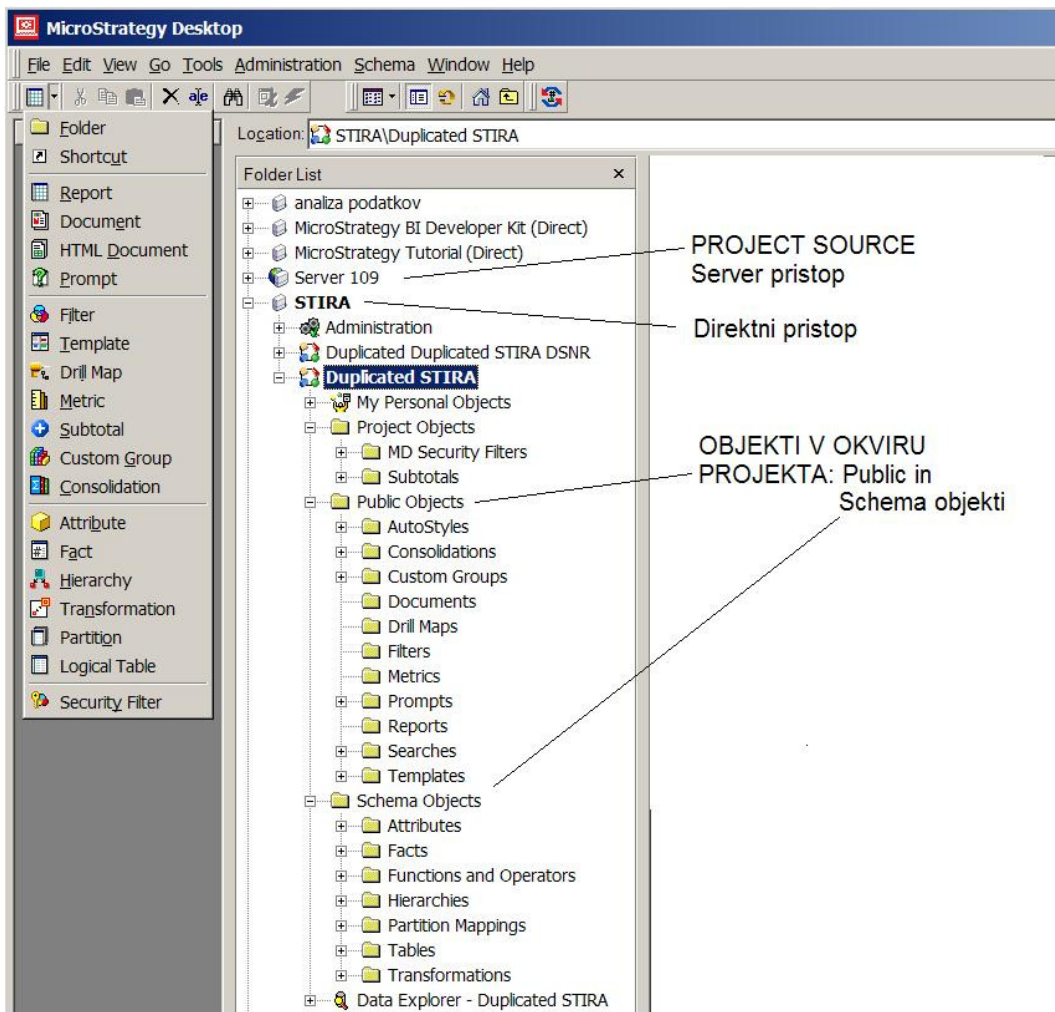
Pri implementaciji projekta STIRA smo se držali korakov, ki smo se jih naučili na tečaju za orodje Microstrategy, in sicer;

1. **kreiranje Project Sourca:** je nadrejeni vir, v okviru katerega kreiramo projekte. Vsakemu Project Sourcu določimo meta bazo. Dostop je lahko direkten ali preko ODBC povezave;
2. **kreiranje Projekta** – v okviru Project Sourca imamo lahko več projektov, zavedati pa se moramo, da so vezani na isto meta bazo. Primer prikaza Project Sourca in Projekta prikazuje Slika 15. Na projektu določimo ODBC povezavo do podatkovnega skladišča. Projektu določimo povezavo do podatkovnega skladišča (angl. *Data Warehouse Catalog*): naložimo podatkovno skladišče oziroma fizično povežemo dimenzijske tabele s projektom;
3. **definiramo objekte v okviru Projekta:** poznamo Schema Objekte, ki jih lahko kreira le razvijalec ter Public objekte, ki jih lahko kreirajo in uporabljajo uporabniki. Do omenjenih objektov lahko dostopamo preko padajočega menija v okviru Projekta ali preko bližnjic iz menijske vrstice;
4. **definiramo attribute:** atributi so elementi dimenzij in nam povedo, kaj želimo spremljati. Za projekt STIRA so bili definirani atributi: Zdravnik, Izvajalec, Dejavnost, Oznaka Zdravnika ter Datum obdobje;
5. **definiramo metrike:** so večinoma številčne vrednosti, ki se nahajajo v tabeli dejstev. Definirane so bile naslednje vrednosti: Število receptov, Število opredeljenih oseb, Število pogojnih opredeljenih oseb, Vrednost receptov, Ciljna vrednost ter Indeks;
6. **definiramo hierarhije:** določimo podrejenost in nadrejenost atributov (Slika 16). Izvajalec ima lahko več dejavnosti in zaposluje več zdravnikov. Prav tako eni dejavnosti pripada več zdravnikov in en zdravnik pripada točno enemu izvajalcu. Ker se en izvajalec lahko pojavi v več dejavnostih in točno določena dejavnost pripada različnim izvajalcem, moramo na atributih določiti povezavo m:m (glej Sliko 17). Možne hierarhije so bile še določene med zdravniki in oznako, saj ima lahko en zdravnik več oznak ter za časovno dimenzijo;
7. **definiramo dinamične filtre** (angl. *Prompt*): z dinamičnimi filtri uporabnik lahko določi oziroma izbere za katere podatke želi imeti prikazano poročilo. Tako so bili definirani: izbira posameznega zdravnika, izvajalca, dejavnost, število receptov, datum obdobje, vnos indeksa, ki je večji od 100, itd.
8. **definiramo filtre:** filtre lahko uporabniki kreirajo tudi sami, če so Web Professional uporabniki. Na Sliki 18 je prikazan primer kreiranja filtra za indeks, ki je večji od 100;
9. **kreiramo poročilo:** ko imamo vse elemente, ki pripadajo poročilu, se lahko lotimo kreiranja poročila;

Slika 15 prikazuje izgled Microstrategy Desktopa.

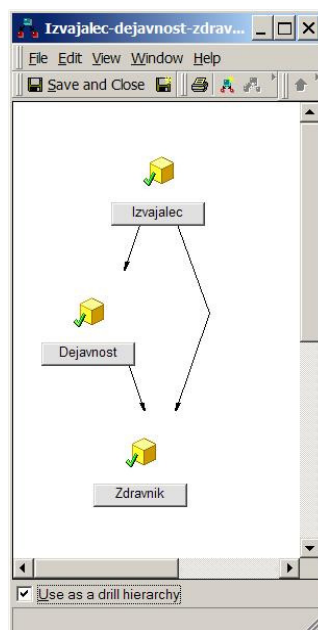


Slika 15: Kreiranje Project Sourca in Projekta ter prikaz objektov projekta



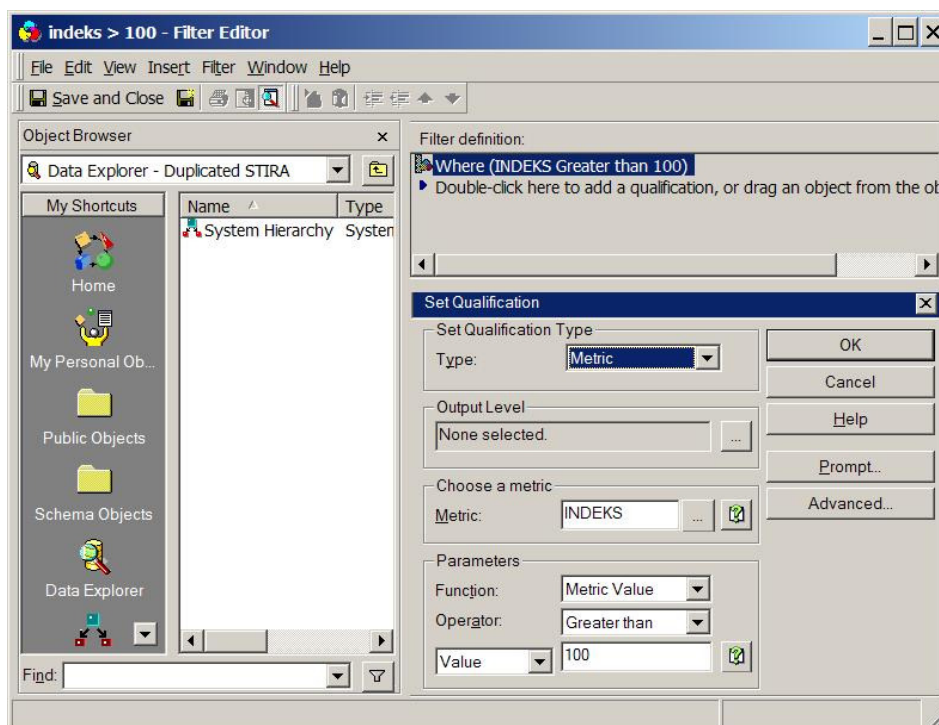
Vir: Orodje Microstrategy, Misrostrategy Desktop, 2008

Slika 16: Primer kreiranja hierarhije



Vir: Orodje Microstrategy, Misrostrategy Desktop, 2008

Slika 17: Kreiranje filtra za indeks > 100



Vir: Orodje Microstrategy, Misrostrategy Desktop, 2008

Glede na zahteve uporabnikov so bila tako kreirana 3 glavna poročila, in sicer:

- prikaz vseh vrednosti za izvajalce;
- prikaz vseh vrednosti za izvajalce in njihove pripadajoče zdravnike po dejavnostih;
- prikaz vrednosti za krovne izvajalce iz agregatne tabele.

Katere podatke vključujejo poročila je že bilo omenjeno.

### 11.3.10 Testiranje delujoče rešitve

Testiranje je bilo najprej naša naloga, da smo se prepričali ali poročila delujejo kot smo si zamislili. Testirali smo iz dveh vidikov, in sicer:

- z namenom, da delujoča rešitev izpolnjuje vse uporabniške zahteve ter
- performančno odzivnost oziroma, da uporabnik z določenimi izbirami (filtri, atributi,...) dobi časovno sprejemljivo želeni rezultat.

Naslednji korak je bil, da smo predali delujočo rešitev ključnim uporabnikom. To so bili v začetni fazi trije oziroma dva, ki sta aktivno testirala delujočo rešitev. Če uporabniku v začetni fazi predstavimo projekt, ki ne deluje ali je stalno kaj narobe, bo dobil odpor do uporabe, kar si ne želimo. Pred tem smo uporabnike tudi ustrezno izobrazili za uporabo olap orodja Microstrategy. V času testiranja so se uporabnikov na nas obračali z vprašanji, ki smo jih skupaj reševali. Po manjših popravkih načrtovane podatkovne tržnice za projekt STIRA, je bila rešitev primerna za produkcijsko rabo.

### 11.3.11 Dokumentiranje v procesu načrtovanja podatkovnega skladišča

Dobra dokumentacija pri razvoju informacijskih rešitev odraža zrelost organizacije in zavedanje, da je to učinkovita naložba za prihodnost. Tega se zavedajo tudi v Zavodu za zdravstveno zavarovanje Slovenije.

V Zavodu je dokumentacija razdeljena na:

- tehnično dokumentacijo oziroma razvojno ter
- uporabniško dokumentacijo oziroma navodil za uporabnike: uporaba orodja Microstrategy.

Za primer projekta STIRA je bilo potrebno napisati za tehnično dokumentacijo:

- povzetek arhitekture rešitve: cilji naloge, dimenzijsko modeliranje, strukture tabel, prikaz arhitekture sistema, postopek in opis polnjenja;
- poročilo o uvedbi razvojne naloge: časovna razporeditev aktivnosti: kdaj je bila izvedena razvojno-testna rešitev, kdaj produkcijska. Katera so ključna poročila ter dostop do njih in njihova uporaba;
- potrditev ustreznosti razvojne naloge s strani končnega uporabnika.

Prednost dobre dokumentacije je predvsem:

- vedno lahko pogledamo za nazaj ter se spomnimo potrebnih postopkov in korakov;
- skrajša razvojni cikel podobnih projektov;
- olajša vzdrževanje in možno nadgradnjo;
- olajšamo delo nekemu, ki bo mogoče v prihodnosti nadaljeval naše delo.

Nekatera podjetja še vedno ne posvečajo dovolj pozornosti pisanju dokumentacije. Tudi ni dovolj, da je dokumentacija napisana in oddana. Potreben je tudi določen nadzor, da vsebuje res tisto, kar bomo kasneje potrebovali.

### **11.3.12 Izobraževanje uporabnikov za delo z orodjem Microstrategy**

Rešitev je bila najprej predstavljena trem ključnim uporabnikom, ki so tudi odgovorni za posredovanje podatkov po območnih enotah. Rešitev so testirali in na koncu po manjših popravkih potrdili delujočo rešitev. Lani jeseni je bilo nato izvedeno večje šolanje za uporabnike, ki bodo poročila uporabljali na območnih enotah.

Glavne prednosti podatkovnega skladišča STIRA so bila že omenjena. Omenim naj le, da bodo poročila doprinesla novo dodano vrednost delovanju projekta, saj je mogoče veliko lažje in hitreje brskati in poizvedovati po podatkih. Uporabnik lahko vidi hierarhične prikaze, izbira attribute, mere, določa filtre, izbira način prikaza in podobno.

### **11.3.13 Vzdrževanje delujoče rešitve**

Za vsako razvito aplikacijo, ki jo oddamo v uporabo je potrebno predvideti tudi kasnejše vzdrževanje. Že z dobro dokumentacijo smo zagotovili lažje vzdrževanje. Čeprav se podatkovno skladišče polni kvartalno, mora biti sistem med delovnim časom vedno razpoložljiv, za kar je odgovoren administrator baze podatkov. Microstrategy Intelligent Server mora vedno delovati. Naloge posameznih članov glede vzdrževanja ali nadgradenj ostajajo enake kot pri razvoju. Aplikacija je pri uporabnikih dobro sprejeta in se že pojavljajo nove zahteve za dopolnjevanje podatkovnega skladišča.

## Zaključek

Cena projektov podatkovnega skladiščenja je v primerjavi z ostalimi projekti na področju informatike zelo visoka, zato se moramo načrtovanja podatkovnega skladišča lotiti s premislekom. V magistrski nalogi sem želela poudariti, da se je vsem fazam razvoja od zbiranja zahtev, postavitve arhitekture, načrtovanja rešitve v olap orodju, polnjenju podatkovnega skladišča pa do testiranja, vzdrževanja in dokumentiranja, potrebno posvetiti enako pozornost.

OLAP torej uporabnikom omogoča na izredno fleksibilen in samostojen način dostopati do podatkov. Predpogoj za učinkovito pridobivanje informacij iz podatkov pa je predhodno dobro pripravljen vir podatkov v podatkovni bazi in uporabniku enostavna in razumljiva uporaba orodja. Način ureditve in prikazovanja podatkov je tu izrednega pomena, saj uporabnika ne zanima kaj se s podatki dogaja, ampak kaj mu podatki povedo.

Uporabniku moramo tudi na prijazen način predstaviti orodje in njegovo uporabo, saj je uspeh uvedbe podatkovnega skladišča odvisen tudi od tega, da se ga aktivno uporablja ter omogoča hitrejše in boljše odločanje na poslovnem področju. Pomembno je tudi, da imamo pred uvedbo podatkovnega skladišča tudi podporo najvišjega vodstva.

Pri izdelavi magistrske naloge in pri svojem delu sem se veliko naučila s področja načrtovanja in uvajanja podatkovnih skladišč. Predvsem se mi je zdelo zanimivo področje kombinacije podatkovnega skladišča in podatkovnega rudarjenja ter s tem odkrivanja neznanih vzorcev in povezav med podatki. Glede na to, da je področje zdravstva precej kompleksno ter da se vedno odkrivajo nove povezave, bi bilo mogoče pametno v prihodnosti posebno pozornost nameniti tudi podatkovnemu rudarjenju. V teorijah se že kar nekaj časa govori o omenjenem področju, v praksi pa se podatkovnemu rudarjenju posveča premalo časa.

Glede na to, da nam orodje Microstrategy ponuja tudi uporabo dinamičnih nadzornih plošč, kjer poročila postanejo še bolj živa in interaktivna, bi se lahko dinamične nadzorne plošče koristno uporabile na različnih področjih v Zavodu za analiziranje in pregledovanje po podatkih.

Projekt STIRA je bil uspešno uveden v poslovanje, uporabniki so novi način dela z veseljem sprejeli, saj jim je bilo omogočeno lažje prikazovanje in brskanje po podatkih. Omogočeni so jim bili tudi povsem novi prikazi kot so uporaba raznih filtrov, dodajali in odzemanli so lahko atribute, mere, imeli so možnosti uporabe statističnih kazalcev, uporabe raznih grafičnih prikazov, uporabo hierarhij itd. Tako so se kmalu po zaključku projekta že pojavile nove zahteve po dopolnjevanju dimenzijskega modela.

Iz prakse sem spoznala, da je proces polnjenja pri razvoju podatkovnih skladišč eden pomembnejših in zahtevnejših procesov, saj se včasih pri celotnem razvoju porabi tudi do 70% vsega časa. Proces ETL-a si lahko predstavljamo kot nekakšno kuhinjo in napolnjeno podatkovno skladišče kot končno jed. Torej, kar bomo skuhalo to bomo jedli. S pomočjo kupljenega ETL orodja DataStage v Zavodu in glede na prednosti, ki jih ponuja, se bo proces polnjenja podatkovnih tržnic v prihodnje še izboljšal. Naša naloga bo dobro spoznati orodje DataStage, da ga bomo lahko kar se da najbolje izkoristili.

Ekipa si je nabrala veliko izkušenj, da se lahko spopade z novimi nalogami gradnje podatkovnih tržnic. Upamo, da se bo naše znanje iz omenjenega področja še nadgrajevalo. Pri

vsakem končanem področnem podatkovnem skladišču, kar velja tudi za projekt STIRA, ne smemo zaspati. Tudi po tem nas čaka delo, saj se podatki lahko spremenijo, uporabniki imajo lahko dodatne zahteve, povečuje se količina podatkov, zato moramo tudi stalno bedeti nad pravočasno odzivnostjo sistema.

Zavod kot institucija na nacionalnem nivoju dnevno izmenjuje podatke, izdeluje analize in primerjave zdravstvenih podatkov in prav področna podatkovna skladišča bodo dodala novo uporabno vrednost na področju hitrejšega odločanja.

## 12 Literatura in viri

1. Bajec, M. (2005). *Informacijske tehnologije*. Ljubljana: Fakulteta za računalništvo in informatiko.
2. Ballard, C., Herreman, D., Schau, D., Bell, R. (1998). *Data Modeling Techniques for Data Warehousing*. First Edition. California: IBM Corporation.
3. Berce, J. (2001). Mala kuharica za skladiščenje podatkov. b.k.: Berce s.p.
4. Berce, J. (2004). *Uresničite svoj potencial, Analysis Services v MS SQL Server 2005*, b.k.: Berce s.p.
5. Berry, M. J. A. (2000): *Matering data mining, the art and science of customer relationship mangement*. New York: John Wiley & Sons.
6. Berson, A. et al. (1999). *Building Data Application for CRM*. New York: McGraw-Hill companies.
7. Deblin, M. (2003). *Ascential DataStage, Designer Guide*. B.k. Ascential Software Corporation.
8. Devlin, B. (1997). *Data Warehouse: From Arhitecture to Implementation*. Reading, Massachusetts. USA: Addison Wesley Longman.
9. Ferle, M. (1996a). Izkopavanje podatkov. *Monitor*, (9), 129-132.
10. Ferle, M. (1996b). Kako izboljšati odzivnost informacijskih sistemov. *Monitor* (11), 132-138.
11. Golob, I. & Welzer, T. (2000). *Arhitekture podatkovnih skladišč*. Najdeno 12. novembra 2008 na spletnem naslovu [http://www.drustvo-informatika.si/fileadmin/dsi2001/sekcija\\_a/golob\\_welzer.doc](http://www.drustvo-informatika.si/fileadmin/dsi2001/sekcija_a/golob_welzer.doc)
12. Gray, P., Watson, H. J. (1998), *Decison Support in the Data Warehouse*, USA: Prentice Hall.
13. Hajitnik, T. (2007). *Poslovna inteligenca - Kako vpeljati BI v Podjetje?* Najdeno 15. december 2008 na spletnem naslovu <http://dne.ena.com/prikaziCL.asp?CIID=10903/>
14. Hackney, D. (1997). *Understanding and Implementing Successful Data Marts*. Massachusetts: Addison-Wesley.
15. Hill, G., Kramer, M. (2002). *A Comparison of Business Intelligence Strategies and Platforms, Comparing Microsoft, Oracle, IBM, and Hyperion*, USA: Microsoft Coropration. Najdeno 3. februar 2009 na spletnem naslovu <http://www.informatica.com/>
16. Hudoklin, A. (2004). *Od podatka do rezultata – uvod v poslovno obveščanje*. NT konferenca. Portorož: ADD d.o.o.
17. Huges, S. at al. (2002). *SQL Server DTS*. Boston: New Riders Publishing.
18. IBM (2004). *Ascential DataStage - Enterprise MVS Edition, Server Job Tutorial, Verzion 7.5.1.* b.k.: IBM Corporation.
19. IBM (2004), *Ascential DataStage - Designer Guide, Verzion 7.5.1*. b.k.: IBM Corporation
20. Imhopff, C., Galemno, N. & Geige, G. J. (2003), *Mastering Data Warehouse Design Relational and Dimensional Techniques*, Indianapolis: Wiley Publishing Inc.
21. Informatica, 2006. *Designer Guide - Informatica PowerCenter, Version 8.1.1.*, b.k.: Informatica Corp..

22. Informatica (2008). *My Informatica.com*. Najdeno 15. januar 2008 na spletnem naslovu <https://my-prod.informatica.com/>
23. Inmon, W.H. (1993). *Building the Data Warehouse*. USA: John Wiley & Sons.
24. Jamnik, M. (2007). *Enterprise Dinamic Dashboards*. Skripta iz izobraževanja. Ljubljana: SNT.
25. Jaklič, J. (2002). *Upravljanje in uporaba podatkov*. Ljubljana: Ekonomska fakulteta.
26. Kastelic, S. (1997). *Implementacija in vzdrževanje DSS*. Ljubljana: ICOS d.o.o
27. Kelly, S. (1994). *Data Warehousing*. USA: John Wiley & Sons.
28. Kimball, R. & Ross M (2002). *The Data Warehouse Toolkit, The Complete Guide to Dimensional Modeling*, (Second Edition). New York: John Wiley & Sons.
29. Kimball, R. & Caserta, J. (2004). *The Data Warehouse ETL Toolkit, Practical Techniques for Extracting, Cleaning, Conforming, and Delivering data*. Indiana: Wiley Publishing.
30. Kimball, R. (1996). *The Data Warehouse Lifecycle Toolkit*. USA: John Wiley & Sons.
31. Kimball, R. (1997), *A Dimensional Modeling Manifesto*. Najdeno 5. marec 2008 na spletnem naslovu <http://www.dbmsmag.com/9708d15.html>
32. Kimball, R. (1996). *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*. USA: John Wiley & Sons.
33. Kimball, R. & Ross M (2007), *The Dimensional Modeling in Depth*. Skripta izobraževanja. Amsterdam.
34. Klaus Schulze, P. (2005). *Microstrategy Education*, Zapiski tečaja. Ljubljana: Microstrategy.
35. Kos, B. (1997), *Izgradnja skladišča podatkov*. Kranj: Fakulteta za organizacijske vede.
36. Krajnc, M. (2005): Projektno vodenje v informatiki, *Projektna pisarna*. Najdeno 15. decembra 2008 na spletnem naslova [www.projektnapisarna.com](http://www.projektnapisarna.com).
37. Lahajnar, S. & Rožanec, A. 2000: Izgradnja večdimenzionalnih podatkovnih baz za programske rešitve OLAP, *Zbornik posvetovanja Dnevi slovenske Informatike*, Portorož: Slovensko društvo Informatika. 100 – 105.
38. Lahdenmaki, T. & Leach, M. (2005), *Relational Database Index Design and the Optimizers DB2*. New York: John Wiley & Sons.
39. LANCom. (2008). Podatkovna skladišča. Najdeno 6. novembra 2008 na spletnem naslovu <http://www.lancom.si/resitve/informacijske/poslovne/podatkovna-skladisca/page.html>
40. Microstrategy. (2005). *Education Course Guide - Microstrategy Architect: Project Design*. Virginia: Microstrategy corp.
41. Microstrategy. (2005). *Microstrategy Education Course Guide - Microstrategy Desktop: Reporting Essentials, Advanced reporting*. Virginia: Microstrategy corp.
42. Microstrategy. (2005). *Microstrategy Education Course Guide: Microstrategy Narrowcast Server Administration*, Virginia: Microstrategy corp.
43. Microstrategy, (2005). *Microstrategy Education Course Guide: Microstrategy Intelligent Server Administration*, Virginia: Microstrategy corp.
44. NPS. (2008). Najdeno 5. februar 2008 na spletnem naslovu <http://www.nps.si/>
45. NPS. (2008). Najdeno 5. februar 2008 na spletnem naslovu [http://www.nps.si/index.php?option=com\\_content&task=view&id=142&Itemid=113](http://www.nps.si/index.php?option=com_content&task=view&id=142&Itemid=113)

46. Pavle, B. (2006), Tehnična predstavitev ETL-a DataStage, IBM WebSphere DataStage, Ljubljana: S&T.
47. Perc, K. (2006). *Izgradnja podatkovnega skladišča in razvoj uporabniške rešitve v tehnologiji OLAP*, Kranj: Fakulteta za organizacijske vede.
48. POE, V. (1996). *Building a Data Warehouse for Decision Support*. New York: Prentice Hall.
49. Rajkovič V. (1989). *Je podatkovno skladiščenje del splošne informacijske kulture?* Najdeno 6. januar 2008 na spletnem naslovu <http://www1.fov.uni-mb.si/programiranje/uros/files/BP/DW98.pdf>
50. Ravbar, A. (2003). *Počasi se spreminjajoče dimenzije*, Ljubljana: Ekonomska fakulteta.
51. SoulEuNet (2003). *Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprise*. Najdeno 10. januarja 2009 na spletnem naslovu <http://soleunet.ijs.si/website/html/rdbbackground.html>.
52. SRC. (2006). *Podatkovno modeliranje*. Zapisni tečaj. Bled: SRC.
53. Šmid, J. (2001). *Poslovno obveščanje včeraj in danes za jutri*. Ljubljana: SRC.SI.
54. Thomsen, E. (2002), *OLAP Solutions. Building Multidimensional Information Systems*. (Second Edition). New York: John Wiley & Sons.
55. Thomsen, E. (1997). *OLAP Solutions*. New York: John Wiley & Sons.
56. ZZZS. (2009). *Spletna stran ZZZS-ja*. Najdeno 16. januar 2009 na spletnem naslovu <http://www.zzzs.si/>
57. Wikipedia. (2009). Wikipedia. Najdeno 5. oktobra na naslovu <http://wapedia.mobi/sl/>.

## **Interni viri**

1. ZZZS, (1999 - 2002). *Dokumentacija direktorskega informacijskega sistema (DIS)*. Interna dokumentacija ZZZS. ZZZS – oddelek za razvoj.
2. ZZZS, (2007). *Dokumentacija o informacijskem sistemu Zavoda za zdravstveno zavarovanje Slovenije*. Interna dokumentacija. Ljubljana: ZZZS - oddelek za razvoj
3. ZZZS. (2007). *Predpisovanje zdravil na recepte po izvajalcih po metodologiji STIRA*. Interna dokumentacija. Ljubljana: ZZZS.
4. ZZZS. (2008). *Zapisniki sestankov projekta STIRA glede arhitekture, uvajanja in vsebine podatkovnih skladišč*. Interna dokumentacija. Ljubljana: ZZZS – oddelek za razvoj
5. ZZZS. (2008). *Dokumentacija projekta STIRA*. Razvojna in produkcijska dokumentacija. Interna dokumentacija. Ljubljana: ZZZS – oddelek za razvoj.
6. ZZZS. (2008). *Povzetek arhitekture rešitve*. Tehnična dokumentacija projekta STIRA. Interna dokumentacija. Ljubljana: ZZZS – oddelek za razvoj.





## 13 PRILOGE

### 13.1 Struktura posameznih tabel za projekt STIRA

V nadaljevanju je prikazana struktura tabel, ki sestavljajo podatkovno tržnico za primer področje STIRA.

#### PRENOS\_STIRA

IME POLJA	POD.TIP.	DOLŽ.	DEC.	NULLS	OPIS
KLJ_IZVAJALCA	INTEGER	4	0	N	Ključ izvajalca
DEJAVNOST_ZDR	INTEGER	4	0	N	Generični ključ za dejavnost
KLJ_ZDRAVNIKA	INTEGER	4	0	N	Ključ zdravnika
OZNAKA_ZDR	INTEGER	4	0	N	Generični ključ za oznako zdr.
DATUM_OBDOBJE	DATE	4	0	N	Datum obdobja
ST_REC	INTEGER	4	0	N	Število receptov
ST_OPR_OS	INTEGER	4	0	N	Število opredeljenih oseb
ST_POG_OS	INTEGER	4	0	N	Število pogojno opred. oseb
VREDNOST_REC	DECIMAL	15	2	N	Vrednost receptov
CILJNA_VREDNOST	DECIMAL	15	2	N	Ciljna vrednost receptov
INDEKS	DECIMAL	15	2	N	Indeks za zdravnika
SIOPERAT	CHAR	8	0	N	Šifra uporabnika
DCZADSPR	TIMESTAMP	10	0	N	Datum in čas zadnje spremembe

#### F\_STIRA

IME POLJA	POD.TIP.	DOLŽ.	DEC.	NULLS	OPIS
KLJ_IZVAJALCA	INTEGER	4	0	N	Ključ izvajalca
KLJ_DEJ_ZDR	INTEGER	4	0	N	Generični ključ za dejavnost
KLJ_ZDRAVNIKA	INTEGER	4	0	N	Ključ zdravnika
KLJ_OZNAKA_ZDR	INTEGER	4	0	N	Generični ključ za oznako zdr.
DATUM_OBDOBJE	DATE	4	0	N	Datum obdobja
ST_REC	INTEGER	4	0	N	Število receptov
ST_OPR_OS	INTEGER	4	0	N	Število opredeljenih oseb
ST_POG_OS	INTEGER	4	0	N	Število pogojno opred. oseb
VREDNOST_REC	DECIMAL	15	2	N	Vrednost receptov
CILJNA_VREDNOST	DECIMAL	15	2	N	Ciljna vrednost receptov
INDEKS	DECIMAL	15	2	N	Indeks za zdravnika
SIOPERAT	CHAR	8	0	N	Šifra uporabnika
DCZADSPR	TIMESTAMP	10	0	N	Datum in čas zadnje spremembe

#### L\_IZVAJALCIOE

IME POLJA	POD.TIP.	DOLŽ.	DEC.	NULLS	OPIS
KLJ_IZVAJALCA	INTEGER	4	0	N	Ključ izvajalca
NAZIV_IZV1	CHAR	60	0	N	Naziv1 za izvajalca
NAZIV_IZV2	CHAR	60	0	N	Naziv2 za izvajalca
KLJ_IZVAJALCA	INTEGER	4	0	N	Ključ OE izvajalca
SIOPERAT	CHAR	8	0	N	Šifra uporabnika
DCZADSPR	TIMESTAMP	10	0	N	Datum in čas zadnje spremembe

**L\_DEJAVNOST\_ZDR**

IME POLJA	POD.TIP.	DOLŽ.	DEC.	NULLS	OPIS
<b>KLJ_DEJ_ZDR</b>	<b>INTEGER</b>	<b>4</b>	<b>0</b>	<b>N</b>	<b>Generični ključ za dejavnost zdr.</b>
NAZIV_DEJ_ZDR	CHAR	60	0	N	Naziv dejavnosti zdravnika
SIOPERAT	CHAR	8	0	N	Šifra uporabnika
DCZADSPR	TIMESTAMP	10	0	N	Datum in čas zadnje spremembe

**L\_OZNAKA\_ZDR**

IME POLJA	POD.TIP.	DOLŽ.	DEC.	NULLS	OPIS
<b>KLJ_OZNAKA_ZDR</b>	<b>INTEGER</b>	<b>4</b>	<b>0</b>	<b>N</b>	<b>Generični ključ za oznako zdr.</b>
SIMBOL_OZ_ZDR	CHAR	7	0	N	Simbol oznake zdr. (&, +, \$\$, VZ)
NAZIV_OZ_ZDR	CHAR	60	0	N	Naziv oznake zdravnika
SIOPERAT	CHAR	8	0	N	Šifra uporabnika
DCZADSPR	TIMESTAMP	10	0	N	Datum in čas zadnje spremembe

**AGREGATNA TABELA ZA DOLOČENE IZVAJALCE****A\_STIRA\_IZV**

IME POLJA	POD.TIP.	DOLŽ.	DEC.	NULLS	OPIS
<b>KLJ_IZVAJALCA</b>	<b>INTEGER</b>	<b>4</b>	<b>0</b>	<b>N</b>	<b>Ključ izvajalca</b>
<b>DATUM_OBDOBJE</b>	<b>DATE</b>	<b>4</b>	<b>0</b>	<b>N</b>	<b>Datum obdobja</b>
ST_REC	INTEGER	4	0	N	Število receptov
ST_OPR_OS	INTEGER	4	0	N	Število opredeljenih oseb
ST_POG_OS	INTEGER	4	0	N	Število pogojno opred. oseb
VREDNOST_REC	DECIMAL	15	2	N	Vrednost receptov
CILJNA_VREDNOST	DECIMAL	15	2	N	Ciljna vrednost receptov
INDEKS	DECIMAL	15	2	N	Indeks za zdr za obdobje
INDEKS_L	DECIMAL	15	2	N	Indeks za zdravnika letni
SIOPERAT	CHAR	8	0	N	Šifra uporabnika
DCZADSPR	TIMESTAMP	10	0	N	Datum in čas zadnje spremembe

Časovno dimenzijo sestavljajo naslednje tabele:

L\_LETO

L\_KVARTAL

L\_MESEC

L\_DAN

Tabela Leto

<b>L_LETO</b>		
leto_id	Integer(2)	Ključ leta
leto_datum	TimeStamp (0)	Datum leta (timestamp)
leto_st_dni	Integer(2)	Število dni v letu
Preteklo_leto_ID	Integer(2)	Ključ prejšnjega leta

Tabela Kvartal

<b>L_KVARTAL</b>		
kvartal_id	Integer(2)	Ključ kvartala
kvartal_naziv	VarChar(100)	Opis kvartala
leto_id	Integer(2)	Ključ leta
kvartal_datum	TimeStamp (0)	Timestamp za kvartal
kvartal_st_dni	Integer(1)	Število dni v kvartalu
Pret_kvartal_id	Integer(2)	Prejšnji kvartal
Pretl_kvartal_id	Integer(2)	Kvartal v prejšnjem letu

Tabela Mesec

<b>L_MESEC</b>		
mesec_id	Integer(4)	Ključ meseca
mesec_naziv	VarChar(100)	Opis Meseca
mesec_leto	Integer(1)	Mesec v letu
kvartal_id	Integer(2)	Ključ kvartala
leto_id	Integer(2)	Ključ leta
mesec_datum	TimeStamp (0)	Timestamp meseca
mesec_st_dni	Integer(1)	Število dni v mesecu
Pret_mesec_id	Integer(4)	Prejšnji mesec
Pret_kvar_mesec_id	Integer(4)	Ključ meseca v prejšnjem kvartalu
Pret_leto_mesec_id	Integer(4)	Ključ meseca v prejšnjem letu

Tabela Dan

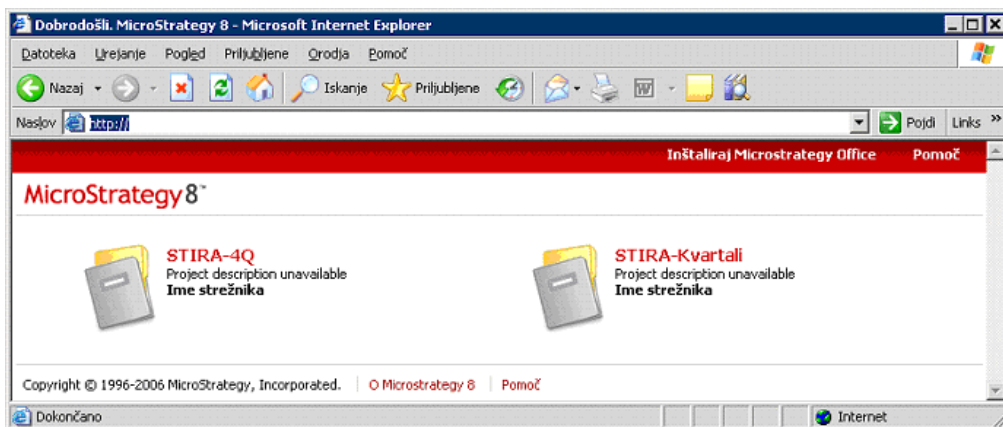
<b>L_DAN</b>		
Dan_id	Date	Ključ dneva
mesec_id	Integer(4)	Ključ meseca
kvartal_id	Integer(2)	Ključ kvartala
leto_id	Integer(2)	Ključ leta

## 13.2 Kratak opis uporabe orodja Microstrategy

### 1 Prijava v sistem in dostop do poročil

Uporabniki za dostop do poročil lahko uporabljajo MicroStrategy desktop ali MicroStrategy Web. Razlika je v tem, da v drugem primeru uporabnik za svoje interaktivne analize uporablja poljuben spletni brskalnik. Funkcionalnosti uporabe poročil in dokumentov so odvisne od pooblastil uporabnika ali licenc. Za uporabnike se priporoča dostop preko web-a, saj je prijaznejši in lažji za uporabo. Uporabnik se v sistem prijavi preko URL naslova ter izbere projekt, ki ga želi uporabljati (Slika 1).

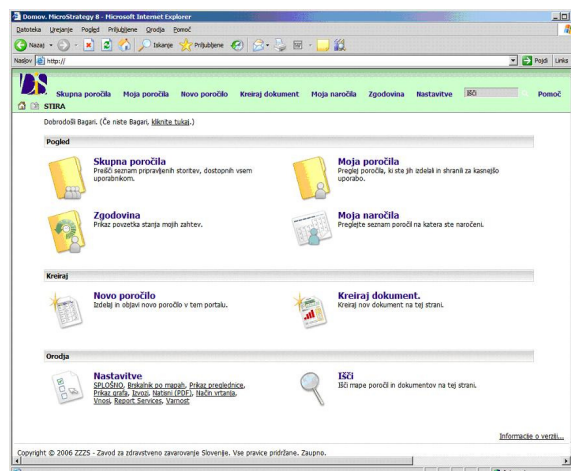
Slika 1: Začetni zaslon ob prijavi



Vir: Orodje Microstrategy, Microstrategy Web, 2008

S klikom na ustrezno ikono se z uporabniškim imenom in geslom prijavimo v sistem. Uporabniku se odpre zaslon z mapami, ki omogočajo pregled in kreiranje poročil (dokumentov) in uporabo orodij za nastavitve in iskanje (Slika 2). V mapi skupna poročila so poročila, ki so dostopna vsem. V mapi Moja poročila so poročila, ki so dostopna samo uporabniku, ki jih je kreiral.

Slika 2: Zaslon za pregled map

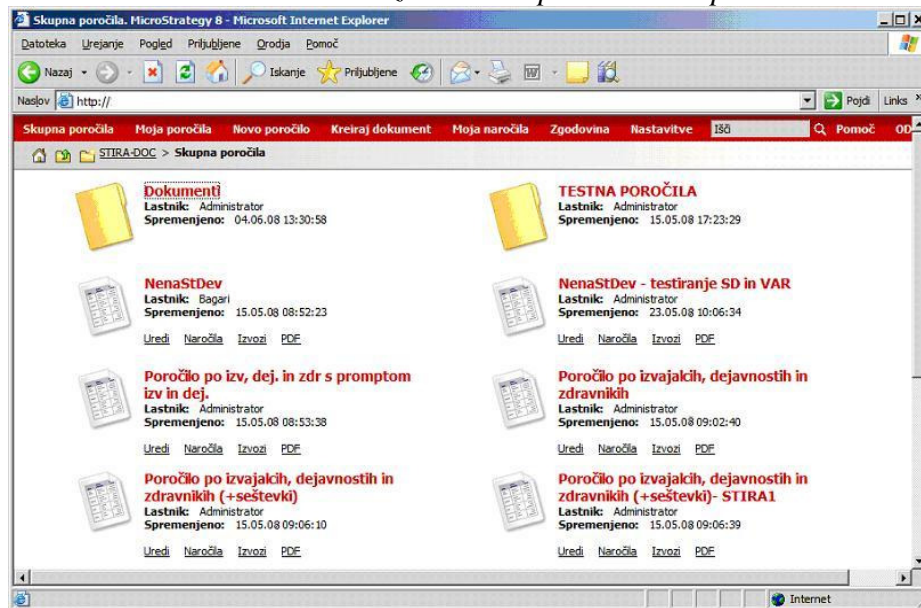


Vir: Orodje Microstrategy, Microstrategy Web, 2008

## 2 Spreminjanje in kreiranje poročil

Uporabnik lahko pregleduje in spreminja poročila. Zato mora imeti ustrezna pooblastila. S klikom na ikono Skupna poročila se prikažejo vsa vnaprej pripravljena poročila ali dokumenti. Primer prikazuje spodnja slika (Slika 3). Če je uporabnik ne dolgo nazaj že poganjal poročilo, se struktura in način dostopa do podatkov shranijo v delovnem spominu (Cache), kar omogoča hitrejše ponovno izvajanje.

Slika 3: Prikaz kreiranih poročil in map



Vir: Orodje Microstrategy, Microstrtegy Web, 2008

Uporabnik za spreminjanje ali kreiranje poročil (dokumentov) uporablja ikone ali opcije menijske vrstice. Primer odprtega poročila prikazuje Slika 4. Podatki so zaradi varovanja zaščite podatkov spremenjeni.

Slika 4: Primer spreminjanja poročila

Izvajalec	Dejavnost	Zdravnik	Kvartal
SPLOŠNA BOLNIŠNICA DR. FRANCA DERGANCA	DERMATOVENEROLOGUA - SEKUNĐARNA DEJAVNO	50 PRAPROTIK FRANC	2006 Q4
		756 BRINUJOC JANEZ	2006 Q4
		<b>Total</b>	
GINEKOLOGUA PRIMARNA DEJAVNOST		35 BERTONCELJ MARJANA	2006 Q4
		4872 BOŠTJANČIČ BOGOMIR	2006 Q4
		6279 NOVAKOVIČ ADOLF	2006 Q4
		675 MASTEN BERTA	2006 Q4
		6867 BRCAR MARLIJA	2006 Q4

Vir: Orodje Microstrategy, Microstrtegy Web, 2008

Zgornja vrstica v glavi poročila nam omogoča naslednje opcije; prehod na mapo skupna poročila, prehod na mapo moja poročila, kreiranje novega poročila in kreiranje novega dokumenta, pregled naročil oz. zahtev, zgodovina poročil, nastavitve (za jezik, pisavo, uro, itd.) in iskanje poročil.

Možnosti glede spreminjanja poročil (dokumentov) oziroma uporabe funkcionalnosti orodja MicroStrategy so razvidne iz spodnjega opisa ikon.

*Tabela 1: Opis ikon v orodju Microstrategy*

Ikone	Opis ikone	Slika ikone	Opis ikone
	Urejevanje pogleda		Združi naslove stolpcev/vrstic
	Shrani		Pogojno oblikovanje
	Tiskaj		Ravnala
	Izreži		Graf
	Prilepi		Preglednica in Graf
	Kopiraj		Poravnaj preglednico
	Undo	<input type="text" value="100%"/>	Povečava
	Redo	<input type="text" value="Tip grafa"/> <input type="text" value="Podtip grafa"/>	Lastnosti grafov
	Izvozi		Legenda
	Briši		Vrednosti podatkov
	Ponovni vnos		Brskalnik objektov
	Osveži		Page-by odsek
	Naroči se	<input type="text" value="Serije:"/>	Lastnosti za grafe
	Izvozi (        )		Uporabi
	PDF oblika		Preglednica / Graf
	Izberi dokument		Pasovi
	Zakleni dokument		Nivojski prikaz
	Tekst		Oblike atributov
	Dodaj sliko		Združi naslove stolpcev
	Črta		Združi naslove vrstic
	Pravokotnik		Zakleni glave stolp.
	Preglednica		Zakleni glave vrstic
	Zamenjaj vrstice s stolpci		Prikaži filter
	Totali/uredi totale		Prilagodi oknu
	Vstavi novo mero		Seznam objektov
	Razvrščanje		združevanje

*Vir: Avtor*

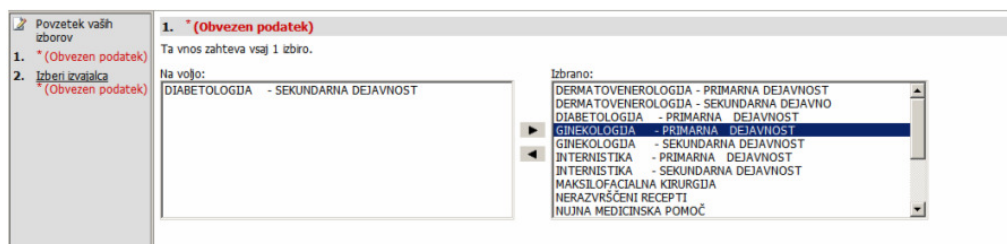
## 2.1 Definicije objektov in pojmov

Poročila v MicroStrategy orodju gradimo iz objektov, kot so:

- **ATRIBUTI** – objekt, po katerem želimo spremljati analize (Npr: dejavnost, zdravniki, izvajalci,...);
- **MERE** – kaj želimo spremljati v okviru atributov (Npr. št. receptov, itd.);
- **FILTRI** – s filtri omejimo prikaz podatkov na poročilu (Npr. določeni izvajalci, dejavnosti,...). Uporabnik ga glede na pravice lahko spreminja ali samo pregleduje;
- **PROMPTI** - so dinamični filtri za določanje omejitev na poročilu;
- **HIERARHIJE** – Določimo podrejenosti in nadrejenosti atributov. Predhodno definirane hierarhije omogočajo drill-anje ali vrtnanje v globino (Npr. vrtnanje z izbiro leta preidemo na kvartale,...);
- **CONSOLIDACIJA** – združevanje objektov (Npr: atribut Poletje = jun + jul + avg);
- **CUSTOM GROUP** – Definiramo določene grupe ali prikazovanje določene spodnje ali zgornje meje vrednosti (npr. Top 10% zdravnikov, ki imajo indeks > 100,...);
- **VRTANJE ALI DRILL-ING** - Možnosti (smer) vrtnanja predhodno določi razvijalec (desktop designer) s pomočjo hierarhij. Uporabniku so tako predstavljene možnosti prikaza podatkov podrejenosti ali nadrejenosti posameznih atributov. (Npr. vrtnanje leta - meseci);
- **AVTOSTYLES** – definiramo stil poročila;
- **TOTALS** – uporabnik si lahko izbere prikaz sumarnih vrednosti po vrsticah ali stolpcih;
- **PAGE-BY ODSEK:** nad podatki poročila imamo možnost, da določen atribut postavimo v PAGE-BY. To pomeni, da se podatki dinamično spremenijo na poročilu, ko izberemo neko vrednost atributa;

V nadaljevanju sta prikazani sliki za primer prompta ali dinamičnega filtra (Slika 5) ter za page-by (Slika 6).

Slika 5: Primer Prompta za poročilo



Vir: Orodje Microstrategy, Microstrtegy Web, 2008

Slika 6: Primer PAGE-BY za poročilo



Vir: Orodje Microstrategy, Microstrtegy Web, 2008



## 2.2 Posebnosti dokumentov

Na dokumentih lahko združujemo več poročil naenkrat. Lahko jih vizuelno preoblikujemo: dodajamo različna ozadja, glavo, nogo, logotip. Dokumenti so tako primerna oblika za tisk ali za posredovanje zunanjim partnerjem. Koristni so, kadar želimo poročila opremiti z različnimi prostimi teksti.

## 3 Zaščita poročil (dokumentov)

Orodje omogoča zelo dobro podporo zaščiti podatkov, da le ti ne pridejo v roke nepooblaščenim osebam.

Omogočena je zaščita na naslednjih ravneh:

- a. zaščita z uporabniškim imenom in geslom: uporabnik mora vnesti omenjena podatka za prijavo in uporabo poročil ali dokumentov projekta;
- b. zaščita na samem projektu: določimo lahko, da imajo do projekta dostop le določeni uporabniki;
- c. zaščita objektov: lahko določimo pooblastilo na nivoju objekta (pregledovanje, spreminjanje);
- d. lahko oblikujemo sezname pooblastil (Security Role): Uporabnike lahko dodelimo različnim grupam uporabnikov. Na nivoju projekta nato povežemo, kateri grupi uporabnikov lahko dodelimo določena pooblastila (npr. pooblastilo za kreiranje novih metrik, dodajanje ali odvzemanje atributov, vrtnanje, kreiranje filtrov);
- e. lahko določimo pravice za uporabo funkcionalnosti (npr. web reporter, web analyst, web profesional). Največ pravic ima web profesional uporabnik, kje lahko kreira tudi posamezne objekte (npr. poročila, mere, filtre, itd), najmanj pa web reporter, ki lahko le poganja predpripravljena poročila).
- f. lahko določimo pravico do raznih map (angl. *folderjev*).

V primeru, da je prijavljenih preveč uporabnikov, nas sistem opozori. Pooblastila, ki jih uporabniki imajo, so omejena s številom zakupljenih licenc na Zavodu.

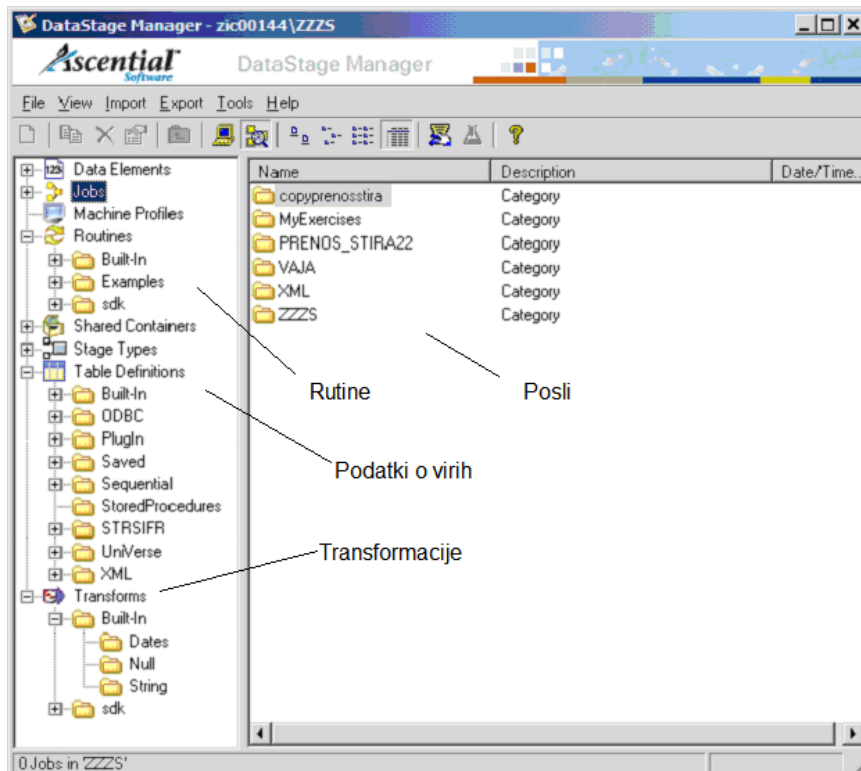
## 13.3 Kratek opis uporabe orodja DataStage na primeru projekta STIRA

### 1 Uporaba orodja DataStage Manager

Je primarni vmesnik do DataStage repozitorija – do definicij tabel, datotek, rutin, transformacij, ki so definirane znotraj projekta.

Omogoča urejanje meta podatkov, ki jih uporabljamo pri gradnji poslov. Je nekakšen vmesnik preko katerega lahko kreiramo lastne transformacije in rutine.

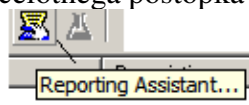
Slika 1a: Prikaz ekranske slike za DataStage Manger



Vir: Orodje DataStage Ascential, DataStage Manager, 2008

Ima tudi močno podporo za dokumentacijo celotnega postopka izvajanja in definiranja posla.

Dokumentacijo kreiramo s klikom na ikono



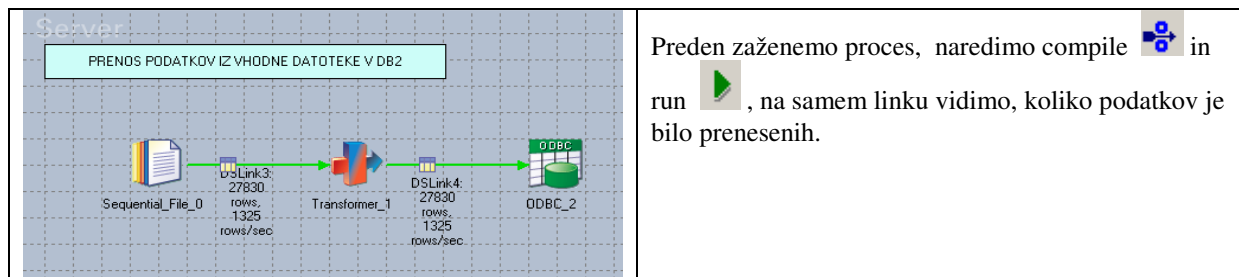
### 2 Uporaba orodja DataStage Designer

DataStage Designer je grafično okolje za izgradnjo posla (Definiranje izvorov in ponorov, definiranje transformacij med podatki, določanje agregacij, združevanje podatkov v več ponorov, definiranje toka podatkov med izvorom in ponorom). Omogoča uporabo objektov za paralelno oziroma vzporedno procesiranje ter pripravo predlog za podobne procese in s hitrejši razvoj (copy/paste).

Lahko se enostavno povežemo na mainframe ali kako drugo bazo na lokalnem ali oddaljenem disku. Definicije tabel, ki jih potrebujemo, lahko uvozimo ali ponovno kreiramo.

Slika 2a prikazuje prenos podatkov iz ASCII datoteke v DB2 bazo.

Slika 2a: Prenos podatkov iz ASCII datoteke v bazo DB2

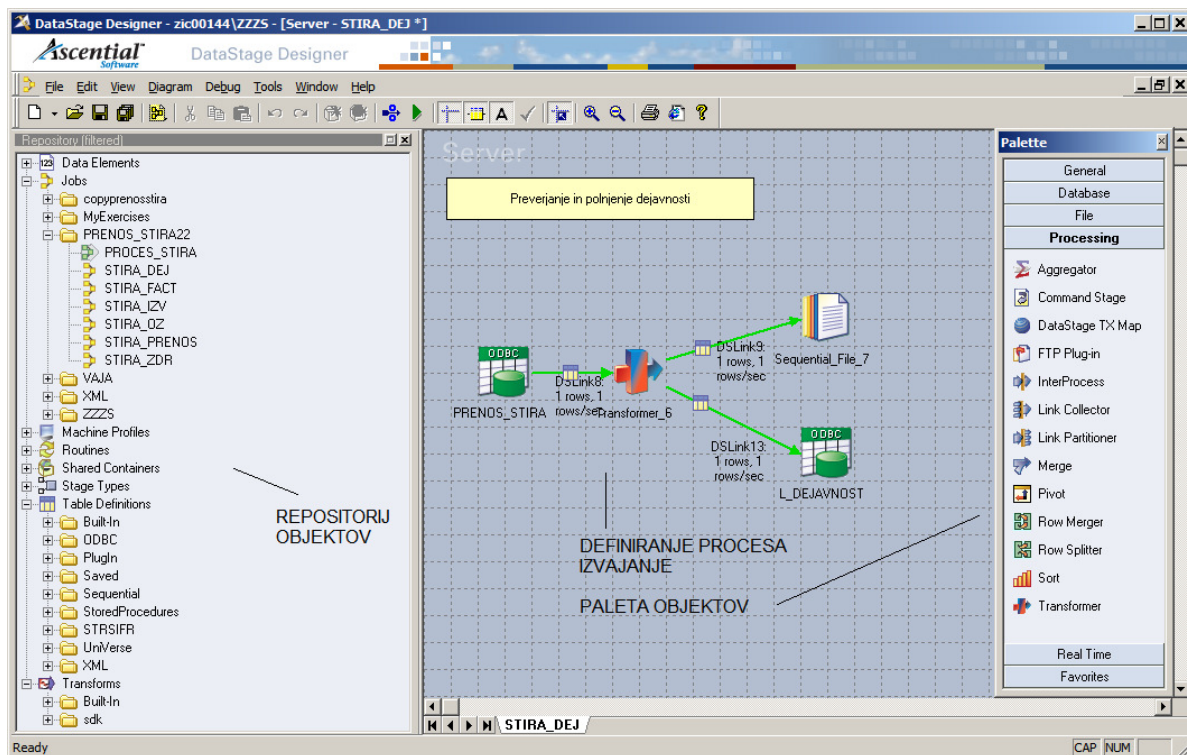


Preden zaženemo proces, naredimo compile in run, na samem linku vidimo, koliko podatkov je bilo prenesenih.

Vir: Orodje DataStage Ascential, DataStage Designer, 2008

Slika 3a prikazuje primer preverjanja in vnosa dejavnosti.

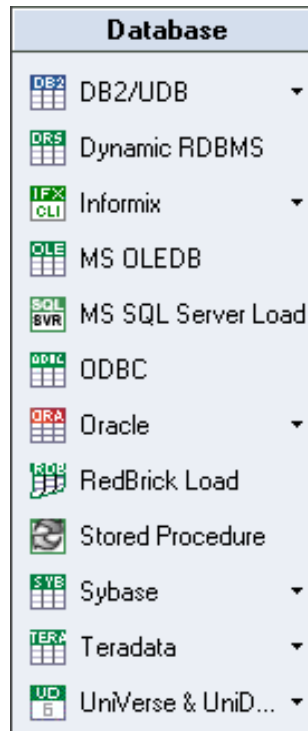
Slika 3a: Zaslonska slika prikaza preverjanja in vnosa dejavnosti z orodjem DataStage Designer



Vir: Orodje DataStage Ascential, DataStage Designer, 2008

Omogočeno nam enostavno testiranje uporabljenih queryjev. Izbiramo lahko med: Generated query, Stored procedure ali User-defined SQL query. Za povezavo do različnih baz, lahko namesto gradnika ODBC uporabimo lahko tudi druge gradnike (Slika 4a).

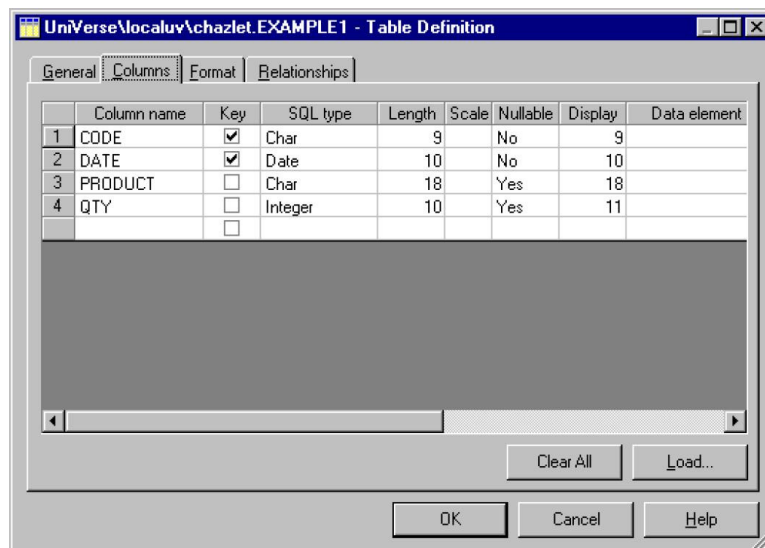
Slika 4a: Gradniki za povezavo do baz



Vir: Orodje DataStage Ascential, DataStage Designer, 2008

Glede na različne gradnike za definiranje baze oziroma tabel, se nekoliko razlikujejo ekrani, ko kliknemo na določen gradnik. Primer gradnika za definiranje stolpcev neke tabele za gradnik UniVerse je prikazan spodaj.

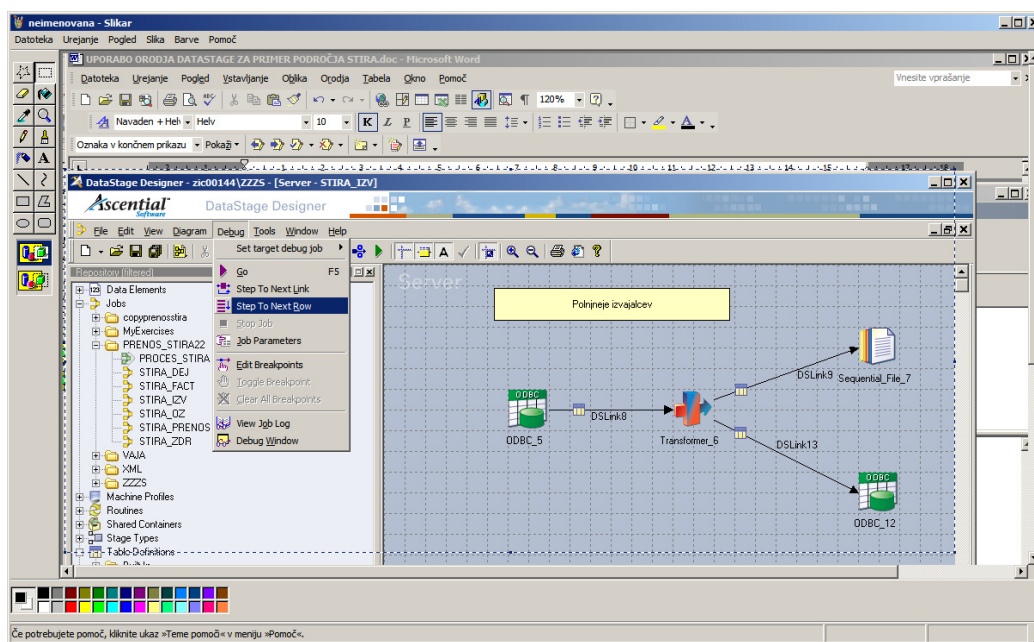
Slika 4b: Primer gradnika za kreiranje stolpcev tabele



Vir: Deblin, M. Ascential DataStage, Designer Guide, 2003, str 42

Orodje nam omogoča tudi dobro podporo pri odkrivanju napak (angl. *debuging*). Primer uporabe je prikazan spodaj (Slika 5a).

Slika 5a : Uporaba opcije Debug



Vir: Orodje DataStage Ascential, DataStage Designer, 2008

### 3 Transformacije in agregacije

Z orodjem DataStage designer (mangerjem) lahko definiramo in uporabljamo lastne transformacije

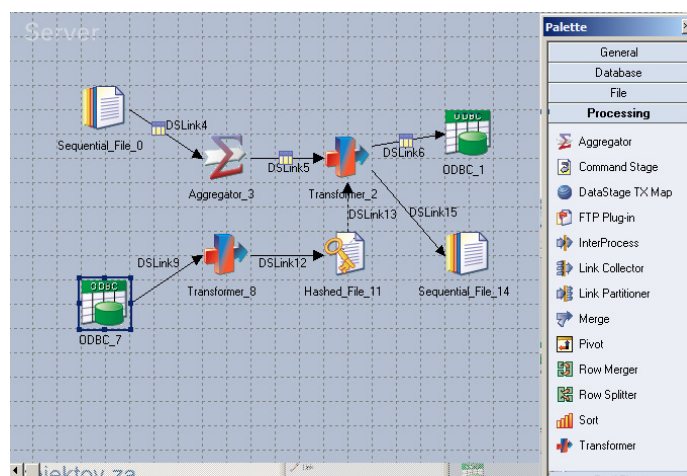
• Osnovne operacije:

Aritmetične (+, -, \*, /), relacijske (=, >, <, <=, >=), logične (AND, OR, NOT), znakovne, IF operator (If then else), Funkcije: TRIM, LEN, UPCASE, DOWNCASE, ICONV, OCONV.

• Sistemske operacije – @DATE, @TIME, @INROWNUM, @OUTROWNUM, @NULL, @WHO, @LOGNAME

Orodje omogoča uporabo gradnikov za razne agregacije in transformacije ter uporabo vgrajenih in uporabniško definiranih funkcij (Slika 6a).

Slika 6a: Uporaba gradnikov za agregacije in transformacije



Vir: Orodje DataStage Ascential, DataStage Designer, 2008


Procese kreiramo v t.i. Server job-u. Procese povežemo v tok zaporedja izvajanja tako, da kreiramo Sequence job (zaporedje tokov). Slika je že bila prikazana v magistrski nalogi pod Sliko 14..

## 4 Uporaba orodja DataStage Director

Možnosti, ki jih nudi DataStage Director:

- zaganjanje poslov;
- priprava časovnih načrtov za posle;
- sledenje izvajanja poslov ter statistike.

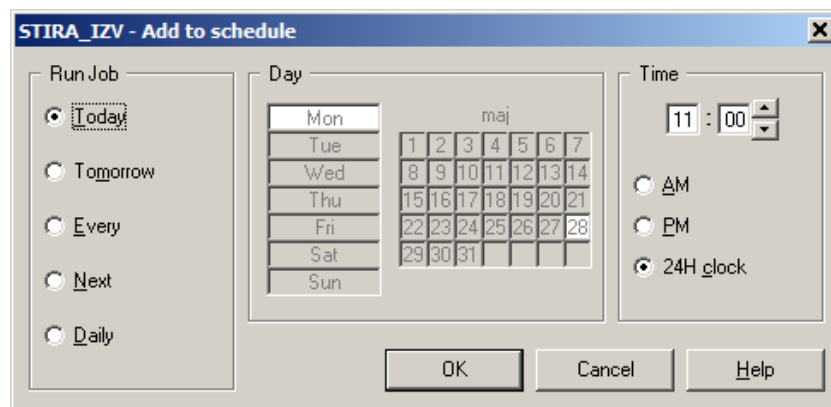
Job lahko prav tako kot v DataStage Designer-ju poženemo s pomočjo DS direktorja. Pri tem lahko pri izvajanju posla opazujemo statuse – angl. *Finished, Compiled, Not compiled,*

*Running*). Pregled dobimo s klikom na ikono  - Status. Vidi se tudi koliko podatkov je bilo prenesenih v določenem času.

Potek izvajanja posla lahko vidimo pod Logi, kjer lahko spremljamo tudi statuse izvajanja. Če je posel zaključen brez napak ima status Finished OK.

Časovno izvajanje posla lahko določimo s klikom na ikono  - Add to Schedule ali izbiro iz menija (Slika 7a).

Slika 7a: Določanje termina za izvajanje posla

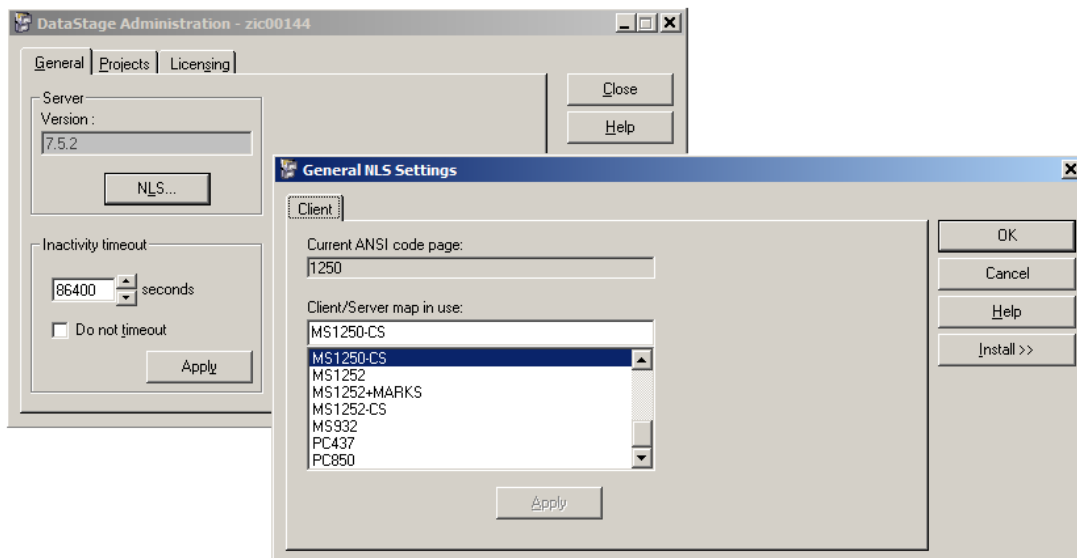


Vir: Orodje DataStage Ascential, DataStage Director, 2008

## 5 Uporaba orodja DataStage Administrator

DataStage Administrator omogoča administracijo projektov in nastavitve DataStage strežnika, projekta (ov). Primer nastavitve oziroma informacij prikazuje Slika 8a.

Slika 8a : Uporaba orodja DataStage Administratorja



Vir: Orodje DataStage Ascential, DataStage Director, 2008

## 14.4 SQL-i za preverjanje in polnjenje podatkovnega skladišča STIRA

- preverjanje zdravnikov

```
SELECT DISTINCT b.KLJ_ZDRAVNIKA FROM prenos_stira B
WHERE b.KLJ_ZDRAVNIKA NOT IN (SELECT a.KLJ_ZDRAVNIKA FROM l_zdravniki
a)
```

- vnosi zdravnikov, ki ne obstajajo nazivi

```
INSERT INTO DB2ADMIN.L_ZDRAVNIKI
SELECT DISTINCT b.KLJ_ZDRAVNIKA, 'NEZNAN','NEDOLOČEN ZDRAVNIK'
FROM prenos_stira B
WHERE b.KLJ_ZDRAVNIKA NOT IN (SELECT a.KLJ_ZDRAVNIKA FROM l_zdravniki
a)
```

- polnjenje tabele L\_IZVAJALCIOE

```
SELECT KLJ_IZVAJALCA FROM l_izvajalci
WHERE KLJ_IZVAJALCA NOT IN (select klj_izvajalca from l_izvajalcioe)
```

- preverimo, kateri zapisi so v tabeli izvajalcev in jih vnesemo z OE

```
SELECT DISTINCT A.OE, A.KLJ_IZVAJALCA, B.NAZIV_IZV1, B.NAZIV_IZV2 FROM
PRENOS_STIRA A, L_IZVAJALCI B
WHERE A.KLJ_IZVAJALCA = B.KLJ_IZVAJALCA
```

- vse, ki so v fact\_tabeli

```
INSERT INTO L_IZVAJALCIOE
SELECT DISTINCT A.KLJ_IZVAJALCA, B.NAZIV_IZV1, B.NAZIV_IZV2, a.oe FROM
PRENOS_STIRA A, L_IZVAJALCI B
WHERE A.KLJ_IZVAJALCA = B.KLJ_IZVAJALCA
```

- vse, ki niso v fact\_tabeli določimo območno enoto (OE) z 1

```

INSERT INTO L_IZVAJALCIOE
SELECT DISTINCT a.KLJ_IZVAJALCA, NAZIV_IZV1, NAZIV_IZV2, 1 FROM
l_izvajalci a
WHERE A.KLJ_IZVAJALCA NOT IN (select B.klj_izvajalca from prenos_stira B)
SELECT DISTINCT a.KLJ_IZVAJALCA, a.NAZIV_IZV1, a.NAZIV_IZV2 FROM
l_izvajalcioe a
WHERE A.KLJ_IZVAJALCA NOT IN (select B.klj_izvajalca from l_izvajalci B)

```

- *preverimo, če je obstaja nova dejavnost*

```

SELECT A.DEJAVNOST FROM prenos_stira a
WHERE A.DEJAVNOST not IN (SELECT b.naziv_dej_zdr FROM l_dejavnost_zdr b)

```

- *preverimo, če se je pojavila v prenosni tabeli, katera nova oznaka*

```

SELECT A.OZNAKA_ZDR FROM prenos_stira a
WHERE A.OZNAKA_ZDR not IN (SELECT b.SIMBOL_OZ_ZDR FROM
l_OZNAKA_zdr b)

```

- Sprememba prenosne tabele. Tam morajo biti točno takšne vrednosti, ki jih potrebujemo za kasnejše polnjenje fact tabele

```

UPDATE prenos_stira
SET dejavnost = 'NEDOLOČENA DEJAVNOST'
where dejavnost IS NULL

```

```

update prenos_stira
set oznaka_zdr = ''
where oznaka_zdr is null

```

- *Polnjenje fact tabele*

```

INSERT INTO FACT_STIRA
SELECT a.klj_izvajalca, b.klj_dej_zdr, a.klj_zdravnika, c.klj_oznaka_zdr,
a.ST_REC, a.ST_OPR_OS, a.ST_POG_OS, a.VREDNOST_REC, a.CILJNA_VRED,
a.INDEKS_o, A.DATUM_OBDOBJE, a.indeks_l
FROM prenos_stira A, L_DEJAVNOST_ZDR B, L_OZNAKA_ZDR c
WHERE A.DEJAVNOST = B.NAZIV_DEJ_ZDR
AND c.SIMBOL_OZ_ZDR = A.OZNAKA_ZDR AND A.DATUM_OBDOBJE =
'XX.XX.XXXX'

```

- *Preverimo, koliko zapisov je bilo vnesenih. Vrstic mora biti točno toliko kot v prenosni tabeli*

```

SELECT COUNT(*) FROM FACT_STIRA
WHERE DATUM_OBDOBJE = '30.09.2007'
UNION ALL SELECT COUNT(*) FROM FACT_STIRA
WHERE DATUM_OBDOBJE = 'XX.XX.XXXX'

```

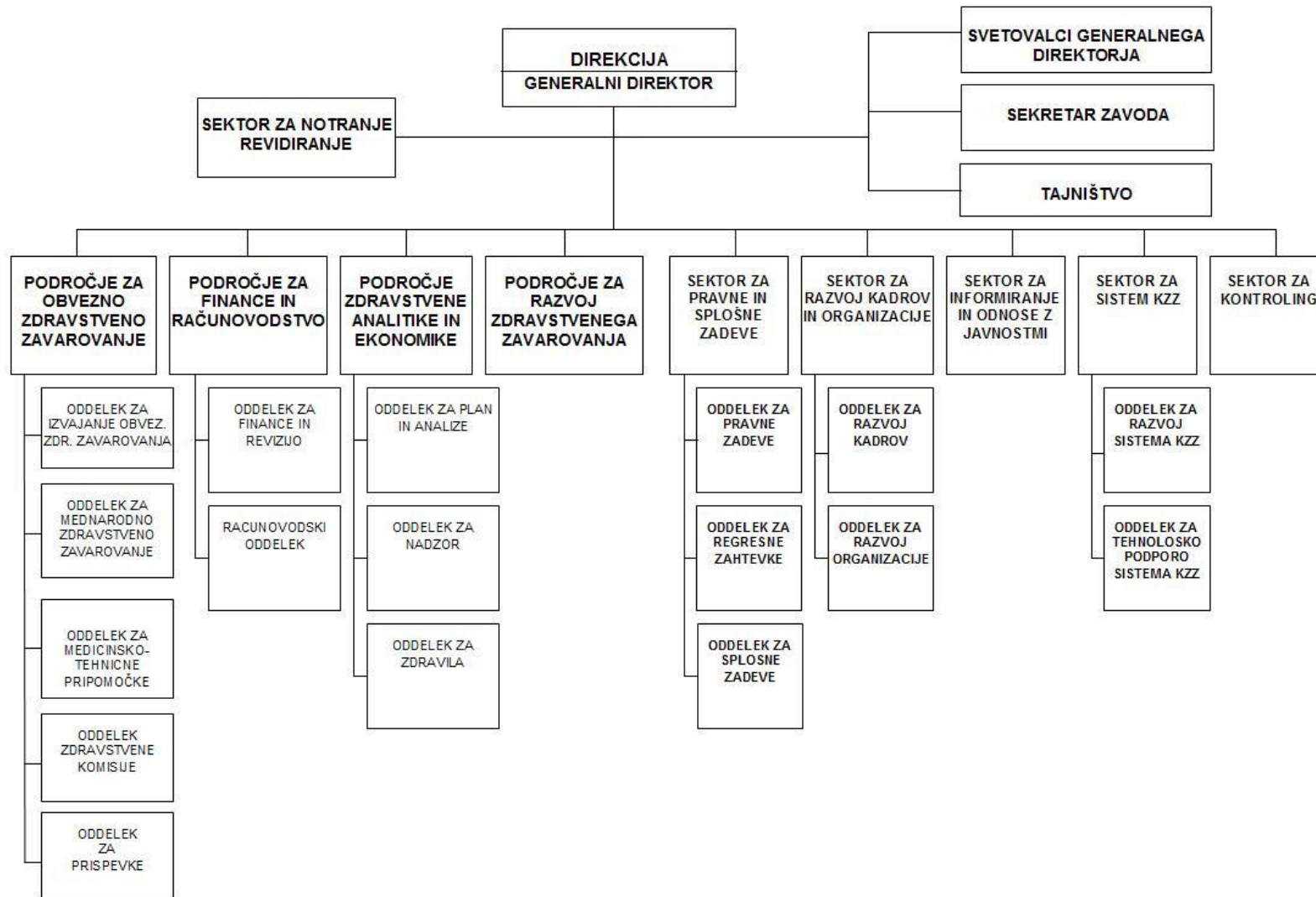
### **13.5 Organigram za direkcijo in poslovno enoto informacijski center.**

Organigram za direkcijo in poslovno enoto informacijski center prikazujeta sliki na naslednji strani.





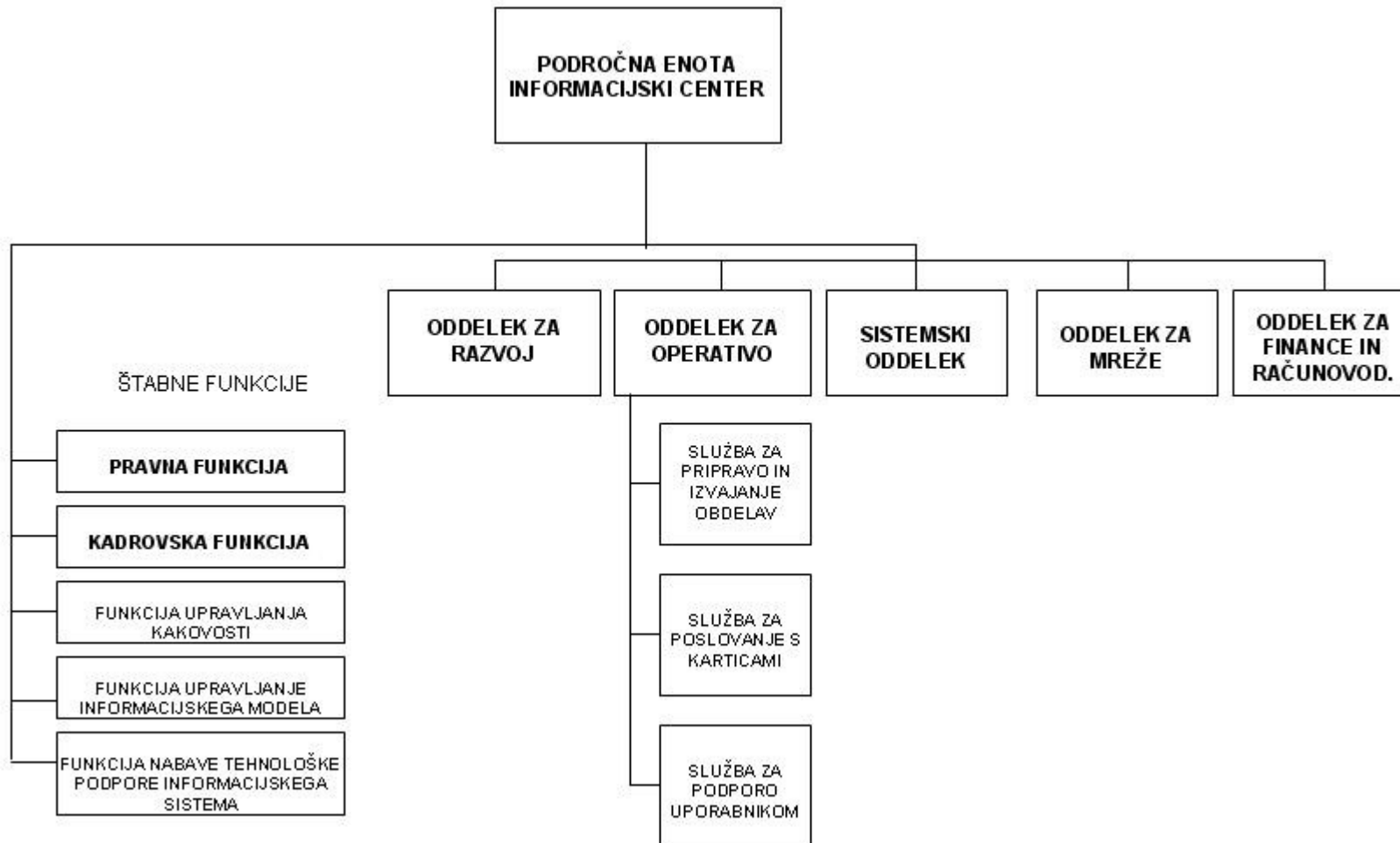
Slika 11: Organigram za direkcijo ZZS



Vir: ZZS, Organizacijska struktura, 2009



Slika 12: Organigram področne enote Informacijski center



Vir: ZZZS, Organizacijska struktura, 2009